MICROCONTROLLORI Sensori capacitivi Display Visione artificiale vasca circuiti stampati Programmazione PIC Pascal PWM PIC18 Dot Matrix RASPBERRY PI MIKROPASCAL

# Fe

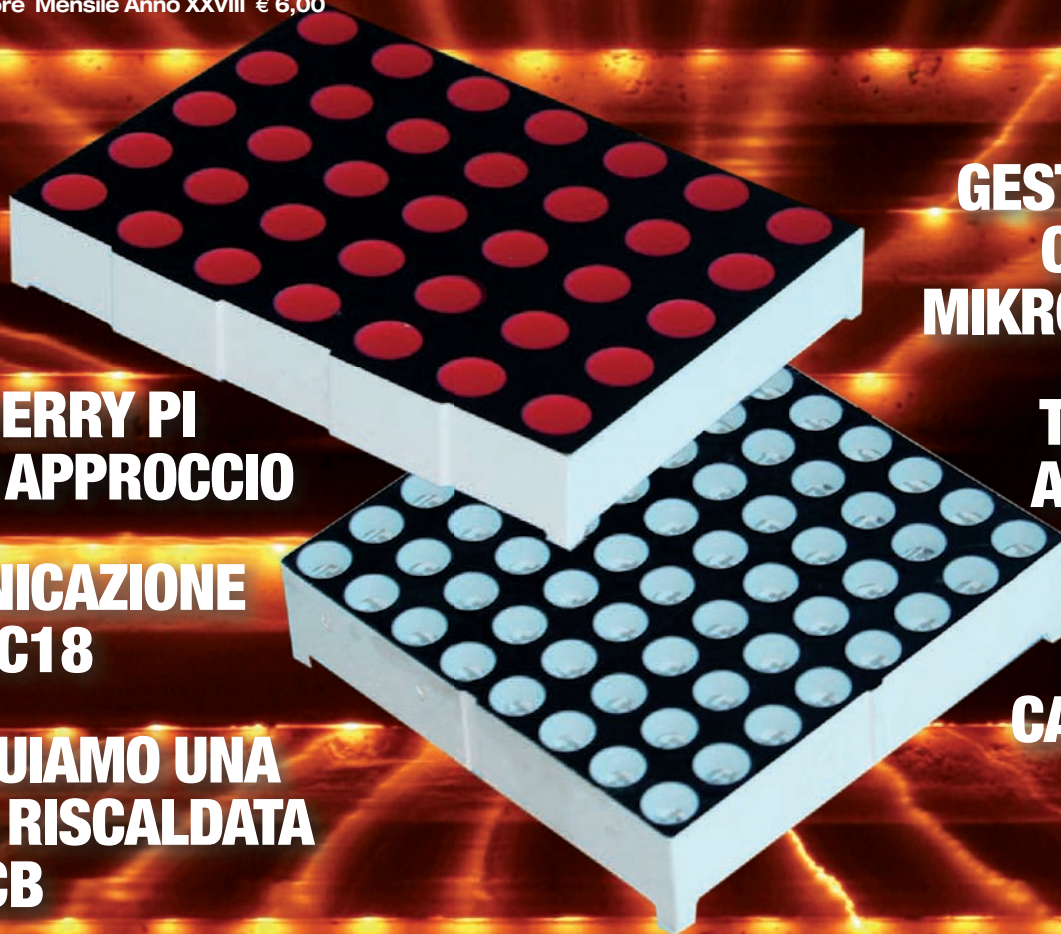
fare elettronica

www.farelettronica.com
329 Novembre Mensile Anno XXVIII € 6,00

# Gestione dei display DOT MATRIX

## GESTIONE I/O CON PIC E MIKROPASCAL

## TECNICHE AVANZATE DI PWM

## SENSORI CAPACITIVI

## RASPBERRY PI PRIMO APPROCCIO

## COMUNICAZIONE CON PIC18

## COSTRUIAMO UNA VASCA RISCALDATA PER PCB

## VISIONE ARTIFICIALE *Identificare oggetti colorati*

ISSN 1591-2272
20329
9 771591 227008

# sommario

**329** *novembre 2012*

## Zoom in


**24** **L'ANTENNA RADIO RAI DI CALTANISSETTA**
Un articolo un po' nostalgico che tratta della storia dell'antenna RAI di Caltanissetta, utilizzata per trasmettere i programmi radio in onde lunghe e medie. Essa è stata per tanto tempo un simbolo della città, in quanto la più alta d'Europa, con i suoi 286 metri di altezza.
*di Giovanni Di Maria*


## Progettare & costruire


**30** **SENSORI CAPACITIVI**
La trasduzione capacitiva è alla base di numerosi sensori industriali, di molti nuovi sensori MEMS e delle moderne interfacce uomo-macchina, come touch-screen, pulsanti e cursori a sfioramento. Analizziamo quindi i princìpi di trasduzione, le diverse configurazioni, gli integrati di misura e le principali applicazioni.
*di Marco Carminati*

**36** **PRIMO APPROCCIO AL RASPBERRY PI**
Tocchiamo con mano il Raspbery Pi e scopriamo assieme le possibilità che ci offre questo piccolo grande gioiello tecnologico dal prezzo estremamente economico.
*di Giovanni Di Maria*

**48** **PWM AVANZATO CON IL PIC (PARTE PRIMA)**
In questa prima parte approfondiamo le conoscenze e le tecniche legate all'utilizzo del segnale PWM di tipo software generato dai microcontrollori PIC.
Nella prossima puntata tratteremo i metodi di generazione di tipo hardware.
*di Giovanni Di Maria*

**56** **VASCA PER PCB HOME MADE**
Realizziamo una vasca per l'incisione di PCB pratica ed economica, in grado di mantenere in temperatura il cloruro ferrico e, al tempo stesso, capace di tenere il fluido in movimento durante le fasi del processo.
*di Gaspare Santaera*


MICROCONTROLLORI Sensori capacitivi Display Visione artificiale vasca circuiti stampati Programmazione PIC Pascal PWM PIC18 Dot Matrix RASPBERRY PI MIKROPASCAL

fe Fare elettronica

Gestione dei display DOT MATRIX

TECNICHE AVANZATE DI PWM

SENSORI CAPACITIVI

GESTIONE I/O CON PIC E MIKROPASCAL

COMUNICAZIONE CON PIC18

COSTRUIAMO UNA VASCA RISCALDATA PER PCB

Esperimenti di Visione Artificiale


*Rispondi* e... **VINCI!** *pag.* **46**

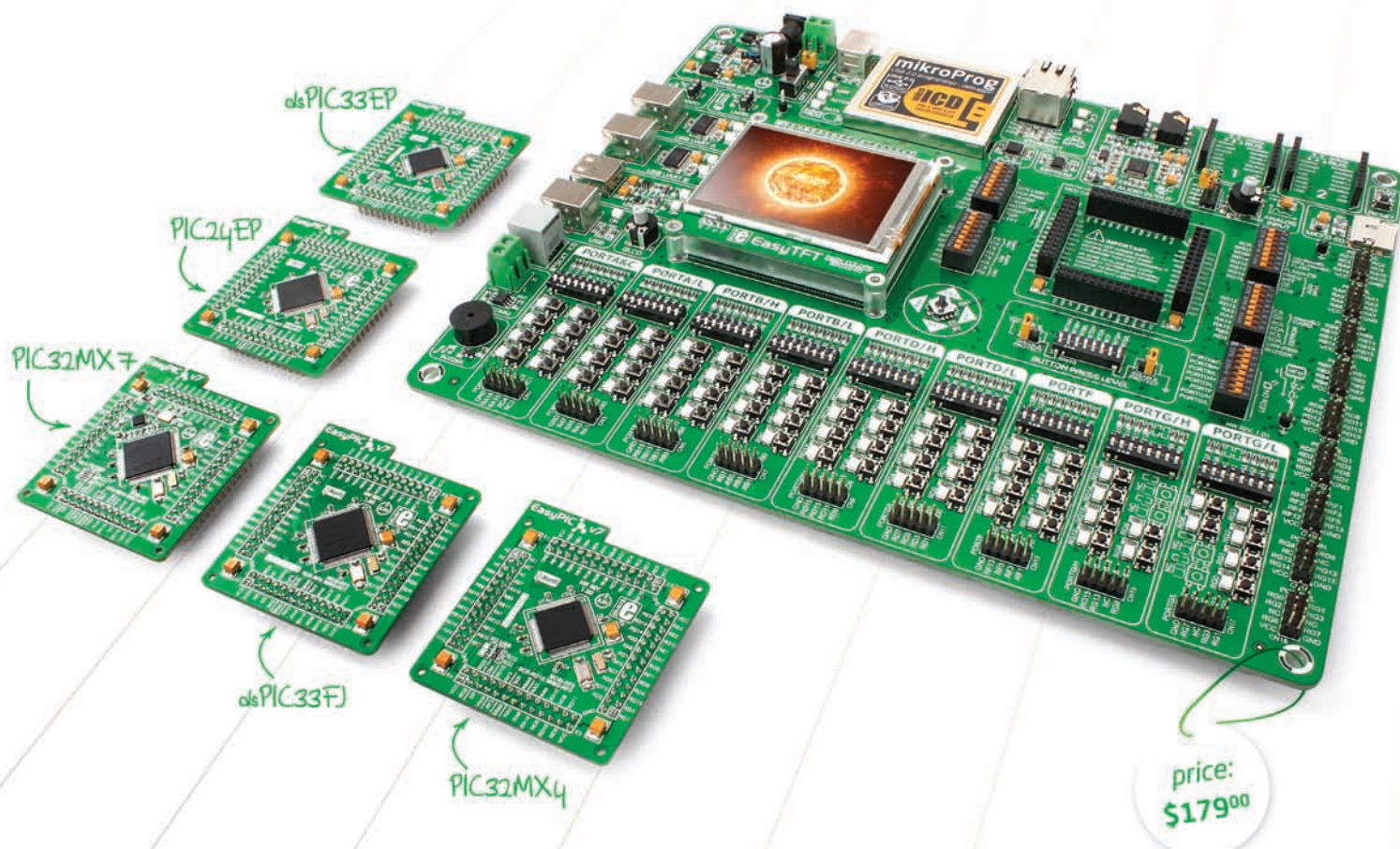
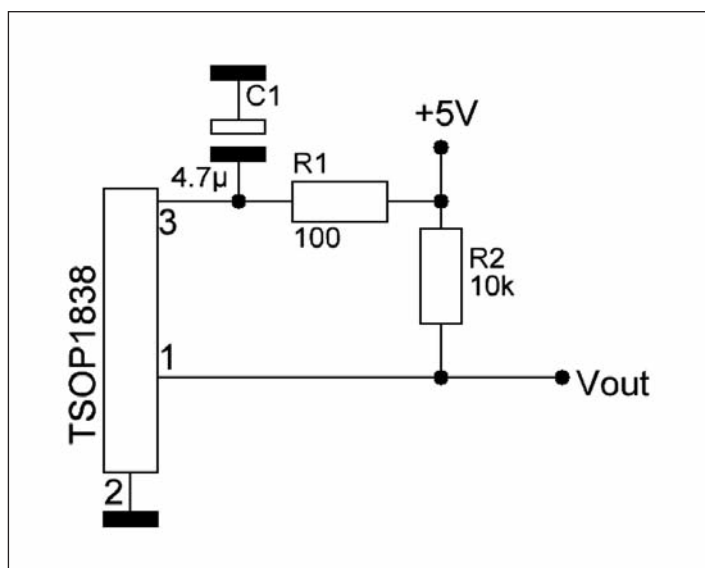

DIVERTITI E METTI ALLA PROVA LE TUE CONOSCENZE CON **ELETTRO QUIZ** E VINCI OGNI MESE ESCLUSIVI PREMI!

di MAURIZIO DEL CORSO

# Fine delle trasmissioni!

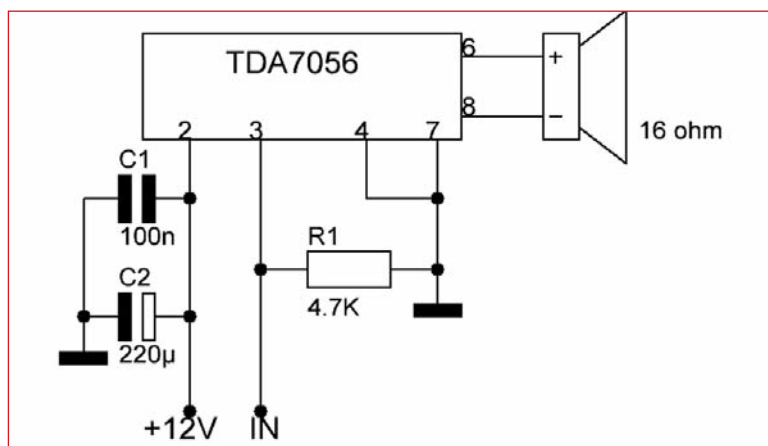
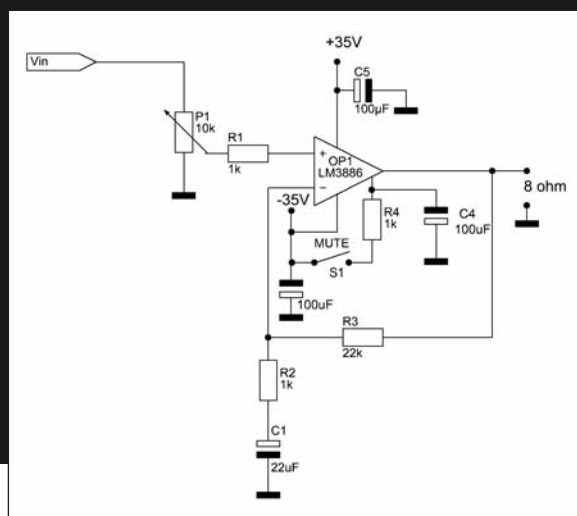
***È imminente lo smantellamento dello storico impianto radiotrasmettitore di Caltanissetta.***

Era il 1924 quando in Italia avvenne ufficialmente la prima trasmissione in radiodiffusione. Dagli archivi Rai è ancora possibile ascoltare l'annuncio dell'allora conduttrice Maria Luisa Boncompagni.
Fu sicuramente una delle prime rivoluzioni tecnologiche. È affascinante ascoltare i nostri nonni che raccontano di quando si riunivano per ascoltare le trasmissioni e posso solo immaginare quale fosse lo spirito con cui si ascoltavano: ognuno era libero di dare un volto a ciascuna voce, immaginarsi i personaggi, i conduttori di cui, solo qualche decina di anni dopo, la TV avrebbe svelato i volti.
Nel 1951 venne costruito l'impianto trasmettitore di Caltanissetta, un'antenna alta oltre 250 metri divenuta un vero e proprio punto di riferimento degli abitanti del posto. Ebbene oggi, visti gli alti costi di manutenzione e visto l'abbandono da parte degli ascoltatori delle frequenze AM, la RAI ha deciso di smantellare questo impianto.
La decisione è stata piuttosto traumatica per gli abitanti del posto e noi di *Fare Elettronica*, tramite la mano del nostro autore Giovanni Di Maria, originario di Caltanissetta, abbiamo voluto rendere omaggio a questa gloriosa antenna che a breve sparirà dall'orizzonte nisseno.
Vi rimando quindi all'articolo di pagina 24 per rivivere in poche pagine la vita e le vicissitudini dell'antenna di Caltanissetta.

di FRANCESCO DI LORENZO

## RICEVITORI IR

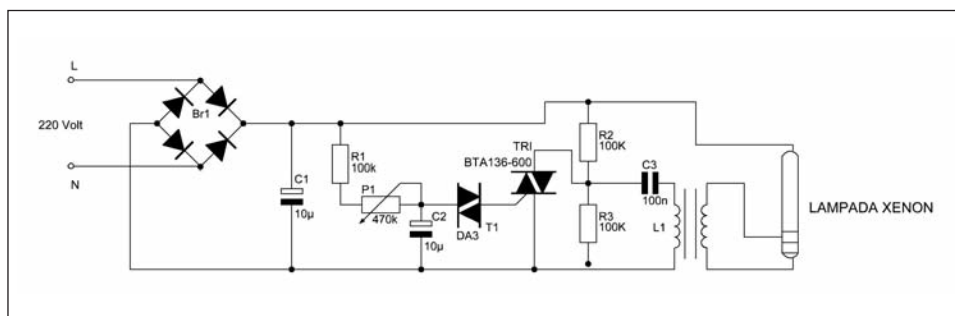
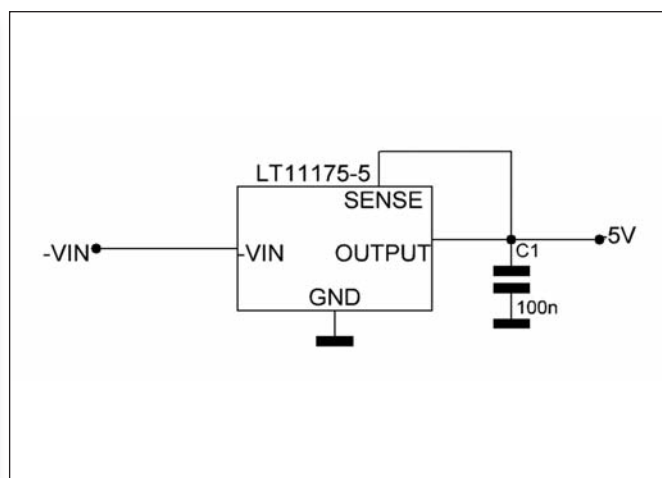
*I TSOP18XX sono una serie miniaturizzata di ricevitori per sistemi di telecomando a raggi infrarossi. Essi incorporano dei preamplificatori. Il segnale in uscita demodulato può essere direttamente decodificato da un microprocessore. Il beneficio principale è l'affidabilità anche in ambienti disturbati e la protezione contro gli impulsi di uscita incontrollati.*

## AMPLIFICATORE DA 68 W

**Questo amplificatore a bassa frequenza viene realizzato mediante un integrato denominato LM3886. È in grado di erogare fino a 60 watt RMS su un carico di 4 ohm o di 50 watt su 8 ohm. Per sfruttare al massimo le prestazioni di tale dispositivo è consigliabile alimentarlo con una duale di 28 V o di 35 V. Questo componente è dotato di un controllo di muto "MUTE", e presenta una distorsione armonica dello 0,03%, una banda passante che va da 20 a 20 KHz, una sensibilità di ingresso di 750 mV e un rapporto segnale rumore di 92dB alla potenza di 1 W.**

## AMPLIFICATORE MONO

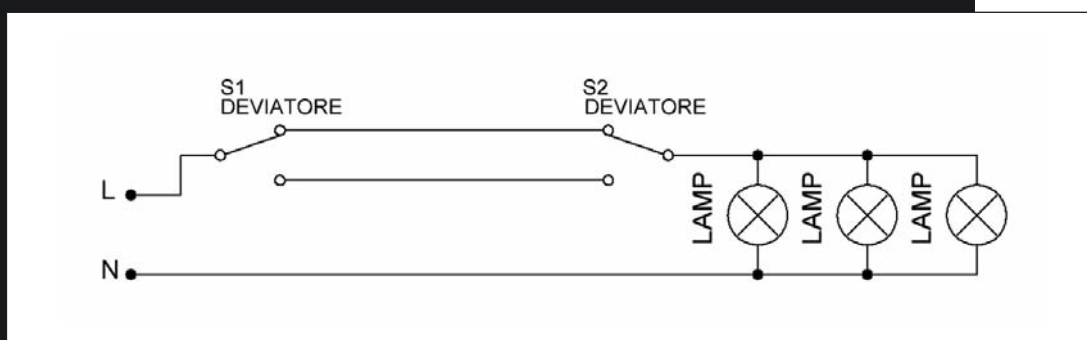
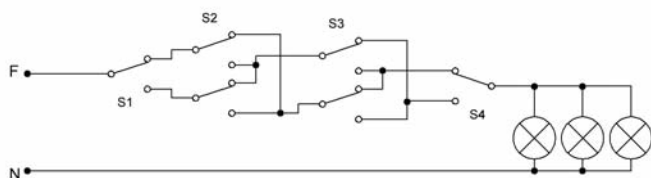
Il TDA7056 è un amplificatore di uscita mono, progettato per essere alimentato da batterie. Essendo stato progettato per applicazioni portatili ha una diminuzione interna dei componenti per evitare spreco di corrente. Esso offre una potenza di 1 W con un altoparlante da 8 W con 6 V di alimentazione o 3 W in un altoparlante da 16 W a 11 V di alimentazione, senza necessità di un dissipatore esterno. Il guadagno è internamente fissato a 40 dB.

## Regolatore low-dropout

*L'LT1117 è un regolatore a basso dropout progettato per fornire fino a 800 mA di corrente d'uscita. Il dispositivo ha diverse tensioni d'uscita, disponibile anche in versioni regolabili e fisse. La versione è stata progettata a 2,85 V specificamente per essere usato in terminatori attivi bus SCSI. Tutti i circuiti interni sono progettati per funzionare fino a 1 V di uscita. La tensione di dropout è garantita a un massimo di 1,2 V a 800 mA.*

## LAMPEGGIATORE STROBOSCOPICO

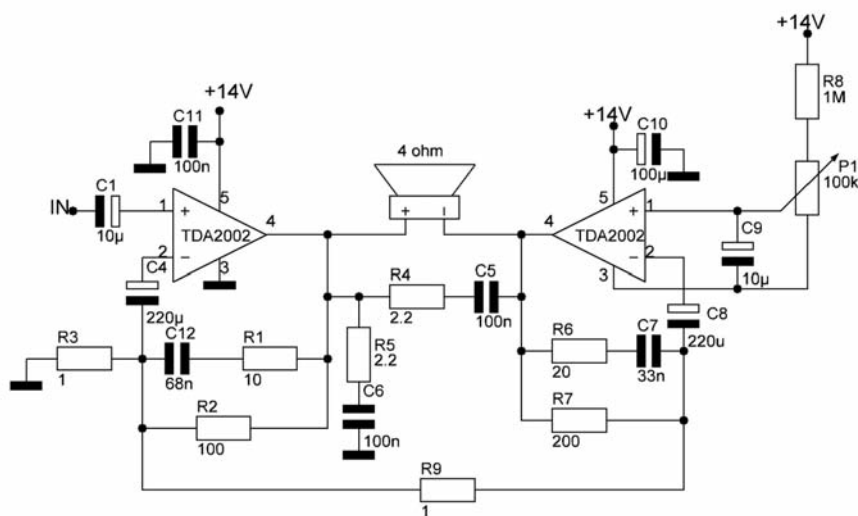
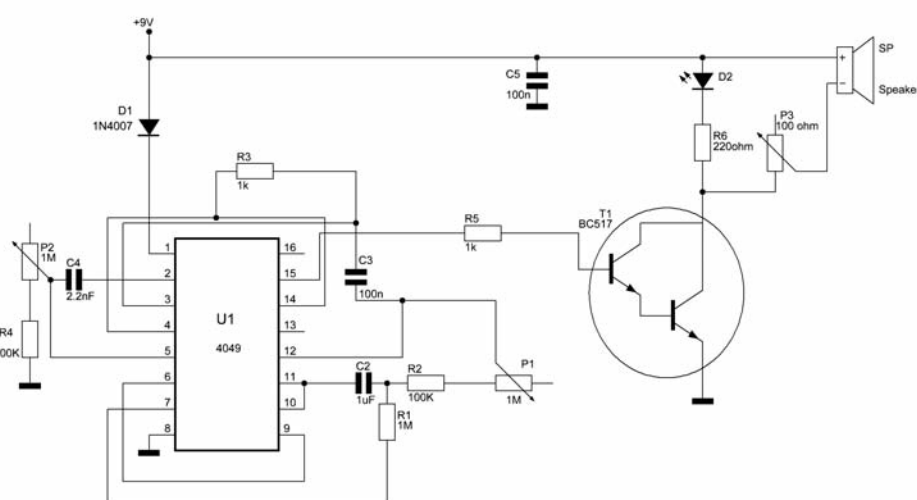
**Genera brevi e intensi lampi di luce bianca usando una lampada allo xenon di piccola potenza. Ideale per lampeggi di emergenza, oppure per visualizzare movimenti in sequenza. È possibile regolare gli impulsi da 2 a 20 al secondo.**

## Comando a due punti

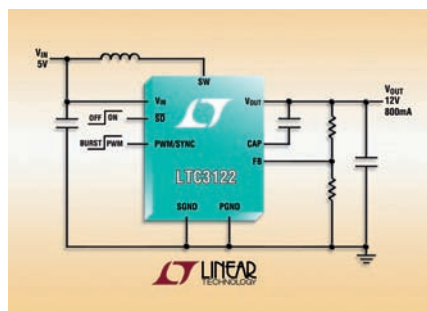
**Questo schema rappresenta, in maniera ideale, come si può realizzare un comando a due punti in un'abitazione. Il circuito è molto semplice da realizzare e si basa sull'utilizzo di due deviatori. Il comando a due punti è ideale per un corridoio.**

## COMANDO A 4 PUNTI

Il comando a quattro punti è utile per abitazioni in cui sono presenti stanze con quattro porte. Il circuito necessita di due deviatori e di due invertitori collegati come in figura. Con lo stesso metodo si possono realizzare comandi a più di quattro punti impiegando due deviatori e un numero variabile di invertitori. Ad esempio il comando a cinque punti si realizza con due deviatori e tre invertitori.

## METRONOMO

Questo metronomo elettronico si basa sull'utilizzo di una porta logica triggerata 4049, da cui vengono creati alcuni oscillatori che danno vita a una serie di commutazioni di 0 e 1. I tre potenziometri consentono di regolare il volume, la timbrica e il tempo. Il circuito va alimentato a 9 V.

## AMPLIFICATORE DA 15 WATT

*Semplice amplificatore audio, piccolo e compatto, molto versatile. Si basa sul TDA 2002, alimentabile fino a 18 volt. Questo tipo di integrato ha avuto un grande successo dagli inizi della sua comparsa, essendo un amplificatore alimentabile a tensioni molto basse. Può essere incorporato in un piccolo contenitore e trasportato ovunque. Per il funzionamento non richiede molti componenti.*

## Regolatore boost sincrono da 2,5 A, 15 V

Linear Technology Corporation presenta l'LTC3122, un convertitore boost DC/DC sincrono a modalità di corrente da 3 MHz con funzionalità integrate di disconnessione delle uscite. Gli switch interni da 2,5 A forniscono tensioni di uscita fino a 15 V da un range di tensioni di ingresso di 1,8 V all'avvio (0,5 V durante il funzionamento) a 5,5 V, rendendo il dispositivo ideale per l'alimentazione a batteria o standard da 3,3 V e 5 V. L'LTC3122 è in grado di fornire una corrente continua in uscita fino a 800 mA, a 12 V da un ingresso a 5 V. Il funzionamento in modalità Burst Mode selezionato tramite pin riduce la corrente di riposo a soli 25 µA e consente una maggiore durata della batteria nelle applicazioni portatili. La combinazione tra package DFN-12 (o MSOP-12) da 3 x 4mm con funzionalità termiche avanzate e frequenza di commutazione costante fino a 3 MHz consente ai progettisti di bilanciare le dimensioni della soluzione PCB con l'efficienza richiesta.

L'LTC3122 utilizza switch interni sincroni con RDS(ON) di soli 121 mOhm (canale N) e 188 mOhm (canale P) in modo da assicurare un rendimento del 95%. La disconnessione delle uscite consente di scaricare completamente la tensione di uscita allo spegnimento ed elimina anche elevate correnti di inrush durante l'avvio, riducendo al minimo la tensione sull'alimentazione. Grazie alla possibilità di regolare la tensione di uscita quando questa è inferiore alla tensione di ingresso, è garantita la compatibilità dell'LTC3122 con qualsiasi tipo di batteria.

**CODICE MIP 2829916**

## Novità nella famiglia Arduino

***Due grandissime novità nella famiglia Arduino: Arduino Due e il nuovo Arduino Starter Kit. Arduino Due è la prima scheda basata su un processore a 32 bit ARM, l'Atmel SAM3X8E ARM Cortex-M3, che migliora tutte le funzionalità standard di Arduino e aggiunge altre nuove caratteristiche. Possiede 54 pin di I/O digitali (16 dei quali possono essere utilizzate come uscite PWM, con risoluzione selezionabile), 12 Pin di Input Analogici 12 bit di risoluzione, 4 UART (porte seriali hardware), e due uscite DAC (digital to analog converter), un cristallo oscillatore a 84 MHz, due connessioni USB, un jack di alimentazione, un header ICSP, un header JTAG e un pulsante di reset. Il nuovo Arduino Starter Kit vi introdurrà nel mondo di Arduino in maniera molto pratica ed è perfetto per imparare e le sperimentare le conoscenze dell'elettronica. Imparerete il facile utilizzo di Arduino mediante la costruzione di diversi progetti creativi. Il kit comprende, oltre alla scheda Arduino UNO rev.3, i componenti più comuni e utili e un libro di oltre 170 pagine con le istruzioni per 15 progetti pratici. Nel kit sono contenuti più componenti rispetto alla richiesta dei progetti pratici proposti, in maniera tale da poter creare qualcosa di personale con le conoscenze acquisite con questo kit.***

**CODICE MIP 2829912**

## GMM 4620: Mini Modulo Microchip PIC18LF4620 di Grifo

Potente ed economico mini modulo basato sul controllore Microchip PIC18LF4680. In un contenitore DIP da 40 piedini, e con 64 K di FLASH, un completo SBC programmabile In Circuit con linguaggi evoluti come C, BASIC, ecc. Il GMM 4620 è in grado di essere utilizzato, come macro componente, direttamente sulla scheda dell'utente. È alimentato a 5 Vdc ed a bordo ha tutto ciò che serve per funzionare e per comunicare tramite una linea seriale, una a livello TTL oppure in RS232. Il grandissimo vantaggio di questo mini modulo è possedere un programma di boot-loader che gli consente di essere programmato/cancellato usando la sola linea di comunicazione seriale. Questo significa che qualsiasi sperimentatore può farsi un circuito, perfettamente funzionante, e programmarlo usando la sola linea seriale. È il componente ideale per risolvere i problemi di automazione sia industriale sia domestica. A questo scopo è sufficiente provvedere, tramite una circuiteria esterna, a bufferizzare le linee di I/O disponibili. Se non si vuole costruire dell'hardware, è possibile utilizzare il GMB HR168, il quale provvede ad alimentare e bufferizzare le linee di I/O del GMM 4620 con 16 ingressi optoisolati, e visualizzati tramite LED, indifferentemente usati come ingressi NPN o PNP, 8 relè da 5 A, una linea di comunicazione bufferizzata CAN, una linea seriale TTL o RS 232, una linea in I2C BUS, RTC con batteria tampone e 240 byte di RAM tamponata. Se invece servono più risorse analogiche, si può ricorrere alla GAB H844 che con le sue 8 linee di A/D converter, 4 Opto-In e 4 relè output diventa una risorsa ideale. Molto interessanti anche le dotazione di software di programmazione, e di esempi, che comprendono vari compilatori C, BASIC tra cui il PIC BASIC Pro Compiler, MicroCode Studio PRO ecc.

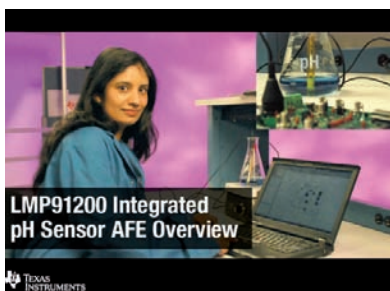
**CODICE MIP 2829490**

## I nuovi controller Microchip PIC a 8 bit combinano alta integrazione, consumi ridotti e rilevamento tattile automatizzato

Microchip annuncia l'ampliamento della sua linea di MCU PIC Enhanced Midrange Core a 8 bit in tecnologia eXtreme low Power (XLP) con la nuova serie di dispositivi PIC16F1512/13. Questi nuovi MCU a 28 pin offrono una potente combinazione di periferiche analogiche e digitali avanzate e, grazie all'adozione della tecnologia XLP, garantiscono l'estensione della vita utile delle batterie imposta da molte applicazioni dell'ultima generazione.
I nuovi MCU, oltre a offrire assorbimenti record sia in modalità attiva (inferiori a 30 µA/MHz) sia in modalità sleep (meno di 20 nA), integrano numerose periferiche, tra le quali un ADC 17x10 bit, una EUSART, un blocco I2C/SPI, due PWM Capture Compare, 7 KB di Flash e 256 B di RAM. Rispetto agli altri modelli della famiglia PIC16F151X, questi due nuovi dispositivi assicurano delle funzionali analogiche superiori grazie alla presenza di un convertitore ADC a 10 bit con supporto CVR destinato all'implementazione di funzioni mTouch di rilevamento tattile capacitivo. Una logica di controllo aggiuntiva consente l'automatizzazione del campionamento di rilevamento tattile, contribuendo così a ridurre le dimensioni del software e a contenere l'uso della CPU. La soluzione offre anche il controllo automatico del pilotaggio degli anelli di guardia e una capacità programmabile di sample-and-hold, assicurando una migliore gestione dei sensori tattili o di prossimità di grandi dimensioni.

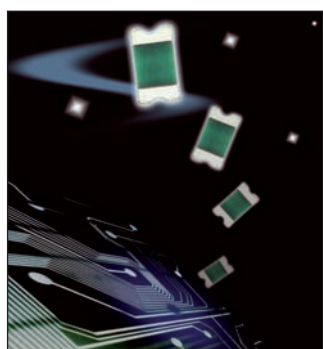
**CODICE MIP 2829312**

## Il primo AFE per il rilevamento di pH al mondo

L'AFE configurabile LMP91200, l'ultima novità della famiglia di soluzioni avanzate per sensori di Texas Instruments, presenta il primo circuito AFE integrato per sensori di pH disponibile sul mercato, in grado di interfacciarsi con tutti gli attuali sensori di pH, facilitando la connessione tra sensore e microprocessore.
Per ottenere misurazioni accurate di pH in settori come il trattamento degli alimenti, la qualità dell'acqua e il trattamento chimico, è necessario superare svariate difficoltà di progettazione che includono ampie variazioni di temperatura, elevate impedenze di uscita, offset e derive. TI contribuisce a superare questi ostacoli nella progettazione di piattaforme per sensori di pH offrendo una soluzione dai consumi ultraridotti che elabora e converte i segnali delle misurazioni con grande precisione, il tutto in un fattore di forma ridotto e integrato.

**CODICE MIP 2829886**

## Toshiba presenta un nuovo contenitore LGA di elevate prestazioni per una linea di MOSFET a bassa RDS(ON)

Toshiba Electronics Europe ha annunciato una serie di MOSFET a bassa resistenza di conduzione che utilizzano il nuovo contenitore UDFN6 a elevate prestazioni. Questo contenitore compatto LGA da 2 x 2 $mm^2$, che garantisce livelli di dissipazione di potenza dell'ordine di 1 watt, permette di gestire correnti elevate in uno spazio limitato, un requisito sempre più indispensabile nei moderni dispositivi mobili.
La serie di MOSFET UDFN6 consiste di dispositivi a canale p o n, singolo o doppio. La gamma a canale p è orientata alle esigenze delle apparecchiature di ricarica e dei sistemi di commutazione del carico, mentre la gamma a canale n è sviluppata in maniera specifica per la protezione della batteria e per i sistemi di ricarica wireless. I MOSFET a canale singolo ammettono una corrente di drain massima di 12 A mentre i MOSFET a canale doppio possono gestire al massimo 4 A. Grazie alla tecnologia UMOS-VI di Toshiba, il nuovo dispositivo SSM6J505NU a canale p singolo presenta una RDS(on) di 61 mΩ (max) con una tensione di commutazione VGS di soli 1,2 V. In base alle esigenze, il progettista può optare per la gamma a canale p, con resistenza di conduzione compresa tra 12 e 95 mΩ (@ Vgs = 4,5 V) e capacità di ingresso compresa tra 290 e 2.700 pF, o per la gamma a canale n, con resistenza di conduzione compresa tra 26 e 64 mΩ (@ Vgs = 4,5 V) e capacità di ingresso compresa tra 270 e 620 pF. Per adattarsi ai diversi usi finali, la tensione di drain nominale massima è pari a 20 V per i dispositivi a canale p e a 30 V per i dispositivi a canale n.

**CODICE MIP 2829261**

## AVX HA SVILUPPATO IL CONDENSATORE RF ORGANICO MULTISTRATO DI ALTA QUALITÀ

AVX Corporation ha sviluppato un condensatore da 500 e 250 VDC, dimensioni di capsula 0603, per montaggio superficiale, in grado di supportare frequenze da 1 MHz e ben oltre 5 GHz. Denominati serie MLO3, tali avanzati condensatori organici RF mantengono alta qualità e alta auto-risonanza, mentre procurano bassi valori di ESR alle alte frequenze. Basati sulla tecnologia brevettata di AVX del multistrato organico a bassa attenuazione (MLO), sono condensatori del tipo organico su base polimerica, che utilizzano interconnessioni in rame ad alta conduttività in modalità multistrato. La capacità di produrre tali componenti su sopporti di ampie superfici con un evoluto sistema di imaging diretto al laser permette migliorati benefici di costo e di controllo delle tolleranze. Il risultato finale è un condensatore RF di basso profilo, basso ESR e alte prestazioni. Inoltre, i condensatori MLO sono corrispondenti per espansione alle schede dei circuiti stampati, al fine di una migliore affidabilità.

**CODICE MIP 2829910**

## I robusti sensori di pressione MEMs garantiscono elevata accuratezza e facilità d'uso grazie a una tecnologia innovativa

Melexis ha presentato il primo di una nuova linea di sensori di pressione MEMs commerciali pronti all'uso in un contenitore standard. Si tratta di un dispositivo estremamente accurato e qualificato secondo le specifiche AEC Q100 per le misure di pressione e particolarmente orientato alle applicazioni da 1 Bar. La particolare tecnologia MEMS piezoresistiva sviluppata da Melexis combina un elemento sensoriale ad elevata accuratezza, un circuito di ingresso analogico a basso rumore e un convertitore analogico-digitale a 16 bit di tipo sigma-delta, che insieme garantiscono prestazioni di livello superiore. La sofisticata catena di trattamento del segnale analogico provvede alla necessaria amplificazione del segnale proveniente dal sensore e alla relativa compensazione dello scostamento (offset). Il microcontrollore a 16 bit integrato si occupa, invece, della compensazione in temperatura e gestisce tutte le funzionalità diagnostiche che sono richieste per garantire il funzionamento in sicurezza delle applicazioni critiche. Una memoria EEPROM integrata completamente programmabile permette di supportare differenti configurazioni, come l'impostazione di funzioni diagnostiche indipendenti per le condizioni di sovra e sottotensione, sovra e sottopressione, così come di personalizzare i filtri digitali per ridurre ulteriormente il rumore di uscita o, in alternativa, per ridurre il tempo di risposta del sensore. La memoria integrata permette anche di registrare i dati di compensazione del sensore e i suoi codici identificativi. Offerto in un robusto contenitore plastico da 16 piedini adatto al montaggio superficiale, il sensore MLX90809 può fornire i dati sulla pressione misurata tramite una tensione di uscita con rapporto proporzionale rispetto alla tensione di alimentazione, oppure mediante il protocollo digitale SENT.

**CODICE MIP 2829908**

## RS Components e Allied Electronics avviano le spedizioni delle schede Raspeberry Pi da 512 MB

*RS Components e Allied Electronics hanno subito iniziato le spedizioni della scheda Raspberry Pi, modello B, nella versione aggiornata a 512 MB, annunciata dalla Fondazione Raspberry Pi.*
*I clienti di tutto il mondo che hanno già un ordine aperto con RS o Allied e sono in attesa della consegna del loro Raspberry Pi da 256 MB, riceveranno automaticamente questa versione aggiornata. Tutti i nuovi ordini effettuati tramite i siti internet RS o Allied saranno portati a compimento con la versione da 512 MB.*
*Questa ultima versione di Raspberry Pi raddoppia la capacità RAM sulla scheda, da 256 MB a 512 MB, ed è prodotta su schede Revision 2, migliorate grazie ai feedback e ai suggerimenti degli utilizzatori, che permettono maggiori performance. I clienti che stanno ordinando tramite RS o Allied beneficeranno di questa funzionalità avanzata senza alcun costo aggiuntivo.*

***CODICE MIP 2829914***

## OMRON E STMICROELECTRONICS PRESENTANO IL D6FG, UN ESCLUSIVO SENSORE PER LA MISURA INTELLIGENTE DEL GAS

OMRON Corporation e STMicroelectronics hanno annunciato il completamento dello sviluppo di un sensore di flusso per gas basato su tecnologia MEMS, che incorpora un esclusivo meccanismo di correzione in base alle differenze nella composizione dei gas, il D6FG. Analogamente a quanto avviene per la misura delle utenze elettriche, anche nella misura delle utenze gas si stanno abbandonando le tradizionali soluzioni meccaniche per passare a contatori elettronici intelligenti dotati di funzioni di lettura automatiche.

Il nuovo D6FG combina un dispositivo Omron all'avanguardia, il trasduttore termico di flusso basato su tecnologia MEMS con un circuito integrato ad alte prestazioni di ST contenente le funzioni di front-end analogico, ottenendo così una misura del flusso di gas molto precisa e ad elevata riproducibilità. I contatori del gas che utilizzeranno il D6FG come soluzione non necessitano di essere configurati per un determinato tipo di gas al momento della spedizione o dell'installazione, poiché sono intrinsecamente capaci di compensare le variazioni di temperatura e di pressione e sono inoltre dotati di un circuito incorporato che compensa le variazioni della composizione dei diversi gas. Il sensore è protetto contro la polvere, per rispettare gli standard internazionali sui contatori del gas.

***CODICE MIP 2829906***

## Fujitsu annuncia un nuovo chip per tag RFID ad alta frequenza con 9 KB di memoria FRAM

**Fujitsu Semiconductor Europe ha arricchito la propria famiglia di chip FerVID per tag RFID con la nuova serie MB89R112 dotata di memoria ferroelettrica (FRAM) che garantisce elevate velocità di scrittura, riscrittura in alta frequenza, tolleranza alle radiazioni e bassi consumi operativi. Con una capacità di memoria di 9 KB, la nuova serie offre soluzioni ideali per l'automazione industriale e i dispositivi medicali, oltre che per le applicazioni industriali ed embedded. Le nuove soluzioni MB89R112 integrano 9 KB di FRAM, la maggiore densità disponibile in un chip RFID operante nella banda HF come definita dallo standard ISO/IEC 15693. 8 KB dei 9 KB disponibili vengono forniti come memoria utente, dando accesso con operazioni di lettura/scrittura all'intera area di 8 KB come previsto dallo standard ISO/IEC 15693. La serie di chip sarà disponibile in due versioni con capacità in ingresso pari rispettivamente a 24 pF e 96 pF. La scrittura di 8 KB di dati richiede circa 4 secondi, operazione che viene compiuta ad altissima velocità, sei volte più rapidamente rispetto ai prodotti E2PROM. I prodotti della linea MB89R112QN aggiungono all'interfaccia RDIF HF una interfaccia seriale SPI supplementare per la connettività verso il microcontrollore.**

***CODICE MIP 2829896***

## Microchip annuncia una nuova famiglia di DSC dsPIC33 "GS"

La famiglia dsPIC33FJ09GS302 rappresenta un ampliamento del portafoglio di DSC dsPIC Microchip e mette a disposizione nuove opzioni a basso costo destinate alla conversione digitale della potenza. I dispositivi della nuova famiglia di DSC integrano, inoltre, delle innovative caratteristiche riducendo contemporaneamente i consumi: ciò consente di ottenere dei livelli superiori di efficienza negli alimentatori AC-DC e DC-DC, nei sistemi di illuminazione HID e LED, negli inverter solari e in altre applicazioni di conversione della potenza. Grazie all'integrazione di ADC ad alta velocità, un core di elaborazione segnali zero-wait-state e di flessibili PWM ad alta risoluzione i cinque elementi della famiglia dsPIC33FJ09GS302 sono ottimizzati per le applicazioni digital-power. L'integrazione di queste periferiche garantisce una interazione ottimizzata che consente di ottenere loop di controllo digitale sub-microsecondo. Tra le altre dotazioni on-chip salienti di questa famiglia spiccano i comparatori analogici con isteresi programmabile e funzionamento rail-to-rail e un modulo PMBus di selezione indirizzabile della sorgente di corrente che permette di ridurre i componenti esterni. Questi nuovi DSC offrono i livelli di dissipazione di potenza più bassi rispetto a qualsiasi altro modello della famiglia GS e sono i primi proposti sia in package SSOP a 20-pin sia in package miniaturizzato VTLA a 36-pin, il quale vanta un ingombro ridotto a soli 5x5 mm. I prodotti della famiglia dsPIC33FJ09GS302 mettono a disposizione l'efficienza di un controllo totalmente digitale per ottenere applicazioni di conversione della potenza con consumi sempre più ridotti e sempre più attente ai costi.

***CODICE MIP 2829502***

## Expoelettronica Forlì

Alla Fiera di Forlì appuntamento con uno dei più importanti eventi nazionali "consumer" di settore: quello con Expo Elettronica. Le cifre sono quelle delle grandi occasioni: circa 400 espositori, 20.000 mq di esposizione e circa 29.000 visitatori attesi alla manifestazione. Expo Elettronica ha un pubblico vasto ed eterogeneo: appassionati del "fai da te", elettro-riparatori, "smanettoni", radioamatori, "cacciatori" di buone occasioni o pezzi rari; questo perché propone un panorama merceologico e un calendario di eventi collaterali veramente ricchissimo. Al suo interno si svolge uno degli appuntamenti più caratteristici e più atteso: il mercatino dell'usato.

Dove: Forlì (FC)
Quando: 1-2 Dicembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: Blunautilus
info: www.expoelettronica.it

**CODICE MIP 2797313**



## SPEZIA EXPO

L'appuntamento per gli appassionati di tecnologia. Espositori qualificati provenienti da tutta Italia proporranno prodotti che spazieranno dai consumabili per PC (DVD e CD vergini, cartucce per stampanti e kit di ricarica), accessori PC, periferiche, navigatori satellitari, lettori multimediali, microcamere, lampade a risparmio energetico, apparecchiature radio, attrezzature fotografiche, telefonia, giocattoli e gadget elettronici e tanto altro ancora. Rispetto ai megastore di distribuzione elettronica, vengono offerti prodotti di grande qualità a prezzi incredibili, è proprio il caso di dire "Grande tecnologia a piccoli prezzi".
In contemporanea a "Elettronica La Spezia" si terrà la "Dolcissima Spezia" DOLCISIMA SPEZIA e al MERCATO DI NATALE.

Dove: La Spezia
Quando: 1-2 Dicembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: Prometeo
info: www.prometeo.tv

**CODICE MIP 2797313**

## Radiantistica Elettronica

***Mostra Mercato Nazionale della Radiantistica Elettronica:***
- ***Materiale radiantistico per C.B. e radioamatori;***
- ***Apparecchiature per telecomunicazioni;***
- ***Surplus;***
- ***Telefonia;***
- ***Computers***
- ***Antenne e Parabole per radioamatori e Tv Sat;***
- ***Radio d'epoca;***
- ***Editoria specializzata.***

***All'interno della manifestazione si terrà il Mercatino del Radioamatore: libero scambio tra privati di apparecchiature amatoriali. In collaborazione con Associazione Radioamatori Italiani - Sezione di Civitanova Marche. Disco: Mostra mercato del disco usato in vinile e CD da collezione.***

***Dove: Civitanova Marche***
***Quando: 8-9 Dicembre 2012***
***Orari: dalle 9.00 alle 18.00***
***Organizzazione: ERF***
***info: www.erf.it***

**CODICE MIP 2797319**

## Mondo Elettronica PIACENZA

***Dopo i notevoli risultati ottenuti nelle fiere di Bologna, Ferrara e Ravenna, Exposition Service organizza un nuovo appuntamento del circuito di fiere dell'elettronica "MONDO ELETTRONICA". La manifestazione si terrà a Piacenza, a ridosso delle festività natalizie.***

***Dove: Piacenza***
***Quando: 15-16 Dicembre 2012***
***Orari: dalle 9.00 alle 18.00***
***Organizzazione: Exposition Service***
***info: www.mondoelettronica.net***

***CODICE MIP 2797323***

## MARC di Genova

Con il Natale alle porte arriva in fiera a Genova il 15 e il 16 dicembre il Marc, una delle migliori occasioni per lo shopping tecnologico. Attrezzature e componentistica radioamatoriale sono il nucleo storico di MARC, alle quali si affiancano e crescono informatica, telefonia, sicurezza, accessori ed editoria specializzata.

Dove: Genova
Quando: 15-16 Dicembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: Studio Fulcro
info: www.studiofulcro.it

***CODICE MIP 2797325***

## Fiera mercato dell'elettronica

Qualità, basso costo dei prodotti venduti e INGRESSO GRATUITO contraddistinguono le fiere organizzate da Printmix. Troverete tutto quello di cui avete bisogno per quanto riguarda l'elettronica: PC, notebook, microcomponenti, cartucce, CD/DVD o accessori. Il tutto a prezzi convenienti, senza trascurare l'ottima qualità che gli espositori presenti sono sempre in grado di offrirvi.

Dove: Mariano Comense (CO)
Quando: 01-02 Dicembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: PrintMix
info: www.printmix.it

***CODICE MIP 2797325***

# L'antenna RADIO RAI DI CALTANISSETTA

Un articolo insolito e un po' nostalgico che tratta della storia dell'antenna RAI di Caltanissetta, utilizzata per trasmettere i programmi radio in onde lunghe e medie. Essa è stata per tanto tempo un simbolo della città, in quanto la più alta d'Europa, con i suoi 286 metri di altezza.

*Figura 1: l'antenna radio RAI di Caltanissetta in una suggestiva vista dal basso.*

Per più di 50 anni l'antenna RAI è stata il simbolo della città di Caltanissetta. Un sorta di torre Eiffel visibile da qualunque luogo anche a una distanza di oltre 30 chilometri. In questi giorni la RAI dovrebbe procedere alla sua dismissione e allo smontaggio definitivo, con grande trepidazione ma anche tanta tristezza. Le persone anziane ricordano con gioia la sua installazione, allorquando i tecnici, veri e propri funamboli, sembravano da lontano tante piccole "formiche" che si arrampicavano sulla colossale struttura.

## L'INSTALLAZIONE

La RAI scelse per l'impianto un luogo strategico, tecnicamente parlando. Un luogo situato al centro della Sicilia che fornisse il proprio segnale radio (almeno quello di terra) a un numero più alto possibile di utenze. I lavori iniziarono del 1949 presso un luogo pianeggiante situato a Caltanissetta, nei pressi della collina di S. Anna. Un posto ottimale per trasmettere un segnale radio ad alta potenza. Un luogo anche molto sicuro, in caso di crollo dell'antenna.
Il sito era controllato 24 ore su 24 da forze dell'ordine che si alternavano continuamente per dare la massima protezione alla struttura.
Nei mesi che seguirono l'antenna diventava sempre più alta, essendo montata pezzo dopo pezzo, e i più anziani ricordano quell'evento con molta gioia e incredulità. Rievocano i lavori con molta lucidità e rammentano che i tecnici che si arrampicavano per i lavori si vedevano a distanza come tanti piccoli puntini mobili, come formiche. Un evento sicuramente eccezionale per quei tempi.
I costi del traliccio e dell'impianto, si dice, ammontarono a più di 50 milioni di lire. La struttura era capace di sopportare il peso di 400 tonnellate. Essa poteva leggermente oscillare alla base, permettendo una leggera elasticità per evitarne la frantumazione. È costituita da un traliccio a sezione quadrata di 3 x 3 metri. La sua inaugurazione avvenne nel 1951.

## LE FREQUENZE

Il traliccio era stato progettato per la trasmissione di programmi RAI nella gamma delle Onde Medie (OM, MW) e delle Onde Lunghe (OL, LW). In onde medie essa ha trasmesso sempre in banda bassa, a 567 kHz, anche per non interferire con il secondo canale RAI, situato a centro banda. Il suo segnale era fortissimo e lo si riceveva praticamente da tutta la Sicilia. Anche chi viaggiava in galleria poteva ascoltare senza problemi le sue trasmissioni. In onde lunghe il traliccio forniva tutta la sua potenza alla frequenza di 189 kHz.

## STRUTTURA FISICA

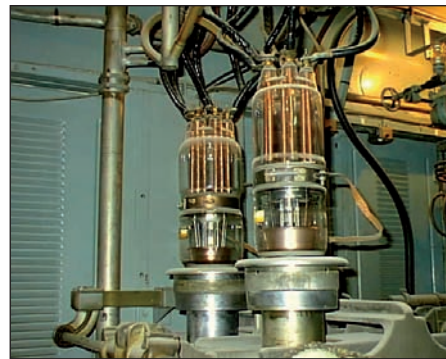
Per la maggior parte dell'attività dell'antenna sono state utilizzate enormi valvole per la trasmissione. Il raffreddamento era ad acqua! Solo dal 1990 esse furono sostituite con i più moderni mosfet di potenza, sicuramente meno romantici.
Il suo principale elemento, come detto, è un'antenna omnidirezionale di 286 metri di altezza. Essa è visibile anche dalla città di Enna, che dista 36 km. Il traliccio è provvisto di robusti tiranti, che la sorreggono in qualsiasi condizione climatica. L'antenna è dotata di luci rosse che si illuminano di notte, segnalando la presenza ad eventuali velivoli.
Da un punto di vista elettrico, l'antenna è

*Figura 2: L'antenna di notte.*

*Figura 3: l'antenna domina sulla città di Caltanissetta.*

*Figura 4, 5 e 6: manutenzione dell'antenna.*

Figura 7: la base basculante dell'antenna.

Figura 8: le maestose valvole trasmittenti da 25 kW.

assolutamente isolata da terra. Eventuali scariche elettriche sono fatte convogliare alla base dove sono installati i cosiddetti "scaricatori". L'antenna è sorretta da due coppie di quattro robusti tiranti, per un totale di otto, periodicamente controllati. Essi sorreggono l'intera struttura con la giusta elasticità dinamica.

## LA DISMISSIONE

Con il passare degli anni, l'attenzione per le onde medie e le onde lunghe è andata pian piano affievolendo. L'avvento di Internet ha inferto un colpo mortale alle trasmissioni, mentre l'interesse si è spostato sulle più comode e performanti onde cortissime e frequenze satellitari. Tutto ciò ha portato alla lenta ma inesorabile inutilità dell'antenna di Caltanissetta, colpevole anche di gravare molto sulle tasche dell'ente radiotelevisivo, per via delle continue e costose manutenzioni ordinarie e straordinarie. Per la manutenzione periodica infatti sono costantemente presenti tecnici specializzati per i lavori ad alta quota, veri e propri funamboli che si arrampicano sulla struttura senza timore. Per questi motivi, le trasmissioni cessarono nel 2004. Restò solo attivo il trasmettitore in onde medie, spento proprio in questi giorni. Ciò che resta adesso è solo un traliccio che dorme, che attende solo lo smontaggio definitivo dalla propria sede, dando l'addio alla popolazione di Caltanissetta che l'ha sempre amata.

## I RICORDI

Qualcuno ha fatto il paragone: Caltanissetta senza l'antenna RAI sarebbe come Parigi senza la torre Eiffel.
L'antenna trasmetteva i palinsesti di Rai Radio 1 (che comunque continuano regolarmente sulla gamma delle F.M.). Iniziava le trasmissioni il famoso "cinguettio" dell'uccellino meccanico.
Il trasmettitore cominciava a irradiare il suo potente segnale alle ore 6:00 del mattino, trasmettendo anche i segnali di "start" alle altre stazioni radio dislocate in tutta la Sicilia e oltre. Tale segnale era caratterizzato dal noto "tic tic tic" seguito da una nota acustica a frequenza fissa della durata di qualche secondo, che si ripeteva anche a mezzanotte. Le trasmissioni con-

Figura 9: visita dell'ARI-PA.

Figura 10: gli uffici della RAI.

Figura 11: la base dell'antenna.

Figura 12: la sala di controllo.

Figura 13: il prof. Michele Fiorito e l'autore dell'articolo Giovanni Di Maria.

Figura 14 sostegni per i tiranti.

tinuavano sino alle ore 24.00, quando la portante si interrompeva, lasciando l'etere libero. Era questa l'ora in cui i radioamatori, gli sperimentatori e i BCL avevano un po' di "libertà" in più. La portante dell'antenna era infatti molto potente e assorbiva in pratica i minimi segnali provenienti da tutto il mondo. Il trasmettitore di Caltanissetta alcuni anni fa mandava in onda, alle ore 15.00, anche un notiziario in lingua araba. Un momento magico che catturava l'attenzione di molte persone che, nonostante non capissero la lingua, restavano incollati all'apparecchio con interesse e curiosità.
Molte persone ricordano ancora il rapporto che si instaurava con i tecnici specializzati che attendevano ai lavori di montaggio dell'antenna. Si trattava di simpatici persone che provenivano dal nord Italia e che trovavano ospitalità presso alcune famiglie siciliane. I tecnici venivano letteralmente "bombardati" di domande che, amabilmente, ricambiavano con altrettante spiegazioni. Sono tante le persone che non vogliono che l'antenna sia rimossa.
Essa ormai fa parte del panorama di Caltanissetta. Molti ricordano i viaggi di rientro verso Caltanissetta nei quali, l'antenna faceva capolino a ogni curva, facendo compagnia durante il viaggio.

Figura 15: il classico panorama di Caltanissetta.

## LE INIZIATIVE E I PROGETTI

Molte sono state le iniziative atte a valorizzare ancor di più l'antenna di Caltanissetta. In passato alcuni proposero addirittura l'allestimento di un grande albero di natale con la sua imponente struttura.
L'iniziativa attuale è, naturalmente, evitare lo smontaggio del traliccio, attraverso una grande opera di persuasione nei confronti della RAI. In effetti, la creazione di un museo darebbe un meritato e glorioso riposo all'antenna, baluardo incontrastato della città.
Chi si distingue alla mobilitazione è l'istituto "S. Mottura" di Caltanissetta che si sta prodigando affinché la RAI desista dall'opera di abbattimento. I due maggiori rappresentanti di questo istituto, il preside Vizzini e il prof. Fiorino, sottolineano che l'aspetto principale dell'impianto non è dato solamente dall'Antenna. Essa è infatti collocata con un particolare sistema chiamato "semisfera su semisfera" ed è sorretta da soli otto tiranti. All'interno dell'area di "Sant'Anna" inoltre trovano sede alcuni trasmettitori di tipo "Marconi", condensatori di altissima fattura, valvole e pensili mantenuti ancora in perfetto stato di conservazione.
Nei magazzini esiste un museo della RAI, con attrezzature ben custodite che sono state utilizzate dagli anni Cinquanta a oggi. Per realizzare un museo di interesse internazionale servirebbe integrare il materiale già esistente con qualcosa che si riferisca ai primi 40 anni, ma che non è assolutamente difficile reperire. L'idea dei nisseni è allestire un museo completo,

Figura 16: i tiranti (evidenziati per maggiore chiarezza).

Figura 20: scala parlante di un vecchio ricevitore.

Figura 17: l'antenna fa ormai parte dell'ambiente nisseno.

destinato a coprire un periodo completo che va dalla nascita di Guglielmo Marconi (1874) a oggi. Da considerare inoltre che l'Antenna di Caltanissetta è stato l'unico ripetitore sul territorio italiano per ben 50 anni.

## CONCLUSIONI

Sarebbe realmente un peccato dismettere ed eliminare completamente l'antenna che ha reso tanto orgogliosi i Siciliani e i nisseni in particolare. Questa antenna ha riempito il cuore di milioni di persone, con i suoi programmi, seppur di minor qualità audio, rispetto alla FM. Per molti l'antenna costituiva un vero conforto, una compagnia, specialmente nelle ore serali. Chi viaggiava in automobile aveva la costante compagnia della radio, in qualunque posto della Sicilia. La frequenza di 567 kHz faceva parte della vita dei siciliani e, con la messa in pensione dell'antenna, è come se un pezzo di quella vita si spegnesse. ❏

**CODICE MIP 2830011**

Figura 18: l'antenna vista dall'alto (fonte: Google Maps).

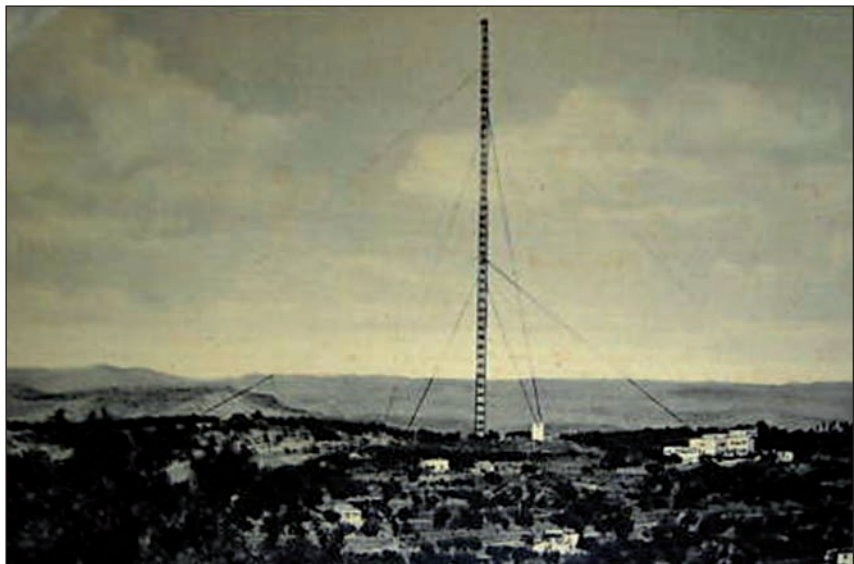



Figura 19: l'antenna raffigurata su una cartolina del 1954.

di MARCO CARMINATI

# SENSORI *capacitivi*

**Grazie all'estrema versatilità e robustezza, la trasduzione capacitiva è alla base di numerosi sensori industriali (ad esempio di distanza, di livello e tomografici), di molti nuovi sensori MEMS e delle moderne interfacce uomo-macchina, come touch-screen, pulsanti e cursori a sfioramento. Analizziamo quindi i princìpi di trasduzione, le diverse configurazioni, gli integrati di misura e le principali applicazioni.**

I punti di forza della misura di capacità, di lunga tradizione in àmbito sensoristico, sono la semplicità e la molteplicità dei circuiti di rilevazione e la robustezza, in particolare offerta dalla possibilità di eseguire misure di quantità variabili nel tempo e nello spazio senza contatto diretto.

## APPLICAZIONI CLASSICHE

Considerando la semplice espressione della capacità esistente tra due elettrodi a facce piane e parallele (di area A e a distanza d): $C = \varepsilon_0 \cdot \varepsilon_r \cdot A/d$ (dove $\varepsilon_0$ è la costante dielettrica del vuoto e $\varepsilon_r$ è la costante dielettrica relativa del mezzo tra le armature) appare evidente come la sua misura possa essere impiegata sia con una geometria fissa (A e d fissi) in cui vari (almeno parzialmente) la $\varepsilon_r$ del mezzo tra gli elettrodi oppure, fissata la $\varepsilon_r$, in cui vari la geometria.
L'esempio più semplice è rappresentata dal sensore di distanza **(figura 1)** in cui un'armatura è fissa e l'altra si muove, come descritto per i sensori MEMS (n. 323 di Fare Elettronica).
Invece, l'applicazione più comune in cui la geometria rimane fissa, mentre cambia la costante dielettrica tra i piatti, è la misura del livello di un liquido in un serbatoio equipaggiato con una coppia di elettrodi disposti in verticale lungo il serbatoio. Via via che il livello del liquido sale, la capacità misurata aumenta, essendo l'aria attraversata dalle linee di campo elettrico tra i due elettrodi sostituita dal liquido con $\varepsilon_r$ maggiore.

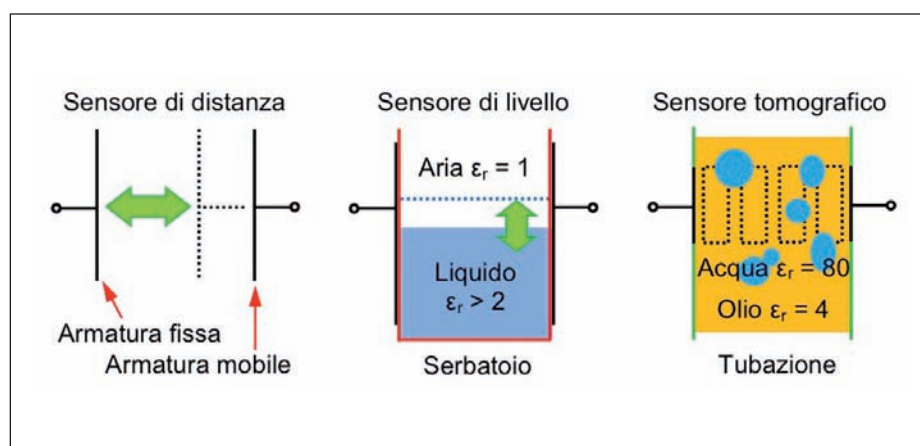
Gli elettrodi possono essere inseriti all'interno del liquido o applicati all'esterno del serbatoio. In quest'ultimo caso, attuabile solo se il serbatoio non è metallico, la sensibilità è leggermente ridotta dalla presenza della parete isolante del serbatoio, ma è preferibile in quanto garantisce una minor usura degli elettrodi, non bagnati dal liquido, ed evita ad esem-



*Figura 1: rassegna di alcune applicazioni significative dei sensori capacitivi: misura dello spostamento, del livello di un liquido in un serbatoio o della sua composizione in un tubo.*

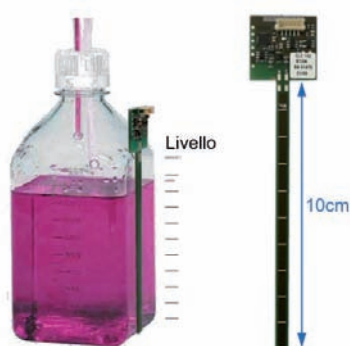



*Figura 2: esempio di un compatto sensore di livello da applicare esternamente a flaconi e serbatoi senza contatto diretto con il liquido.*

pio il rischio di innescare esplosioni con liquidi infiammabili.
Un esempio di questo tipo di sensori è la serie CLC di Sensors Technics **(figura 2)** che offre una risoluzione verticale di 1,5 mm su 10 cm di escursione, con una compatta board di misura con uscita seriale digitale.

## TOMOGRAFIA DI CAPACITÀ

Un'estensione molto interessante della semplice misura di capacità tramite una singola coppia di elettrodi è la tomografia di capacità, che rappresenta un caso particolare della tomografia di impedenza. Questa tecnica consiste nel circondare la superficie di un volume di interesse con un serie di elettrodi e nel ricostruire le proprietà elettriche (dielettriche nel caso della capacità) dei materiali e degli oggetti presenti all'interno del volume ispezionato, combinando opportunamente tutte le possibili permutazioni delle misure dell'impedenza (ad esempio eccitando a rotazione un solo elettrodo e misurando la corrente indotta in tutti gli altri o a coppie).

Maggiore è il numero di elettrodi (e la densità della griglia di misura) e conseguentemente migliore sarà la risoluzione ottenibile, a discapito di una maggiore mole di dati da elaborare e quindi di un'esplosione del peso computazionale. La risoluzione di questo "problema inverso" non è banale dal punto di vista matematico, ma l'ampia diffusione di questa tecnica ha portato alla messa a punto di algoritmi molto efficienti e spesso disponibili gratuitamente, anche se non facilmente migrabili da un'applicazione all'altra.
Le applicazioni spaziano infatti dalla scala delle centinaia di km (come nel caso delle ispezioni elettromagnetiche del suolo alla ricerca di giacimenti petroliferi) a scale di centimetri o millimetri come nel corpo umano (per ricostruire in maniera totalmente non-invasiva, ad esempio, il livello di idratazione di un paziente o monitorare la dilatazione polmonare) o dei condotti in impianti industriali.

Questa tecnica è particolarmente utile quando i materiali da monitorare hanno un elevato contrasto dielettrico (ovvero valori della costante dielettrica molto diversi) come nel caso tipico dei fluidi che scorrono in tubazioni industriali **(figura 1)**. Ad esempio è possibile rilevare tracce di acqua ($\varepsilon_r = 80$) la cui presenza è indesiderata in condutture che trasportano olio ($\varepsilon_r = 2\text{-}4$).
Alcuni esempi della qualità delle immagini "dielettriche" della sezione ottenibili anche con materiali a contrasto relativamente basso (plastica con $\varepsilon_r = 3$ rispetto all'aria con $\varepsilon_r = 1$) con 16 elettrodi attorno ad un tubo di 10 cm di diametro sono illustrati in **figura 3**.
Dal punto di vista esclusivamente elettronico, la complicazione di gestire la rotazione tra un elevato numero di elettrodi si concentra nella progettazione di un adeguato circuito di multiplazione bidirezionale.

## TECNICHE E CIRCUITI INTEGRATI DI MISURA

Un altro punto di forza dei sensori capacitivi è costituito dal fatto che i circuiti di misura possono essere realizzati in moltissimi modi. Le due categorie principali sono la misura nel dominio della frequenza e nel tempo. Nel primo caso una tensione sinusoidale è applicata alla capacità, e la corrente che scorre in essa, letta con un amplificatore a transimpedenza, viene demodulata moltiplicandola per il segnale di riferimento sfasato di 90° o semplicemente raddrizzata.

Alternativamente, la capacità viene inserita in un ponte di Wheatstone capacitivo (o in un semplice partitore raziometrico che rappresenta mezzo ponte) alimentato sempre da un tensione sinusoidale la cui partizione vene letta ai capi della capacità incognita.
Un approccio molto sensibile per misurare piccole variazioni di capacità si basa sulla risonanza. La capacità viene accoppiata a un'induttanza in un tank LC risonante, in modo che le variazioni della frequenza di risonanza riflettano le variazioni di capacità.

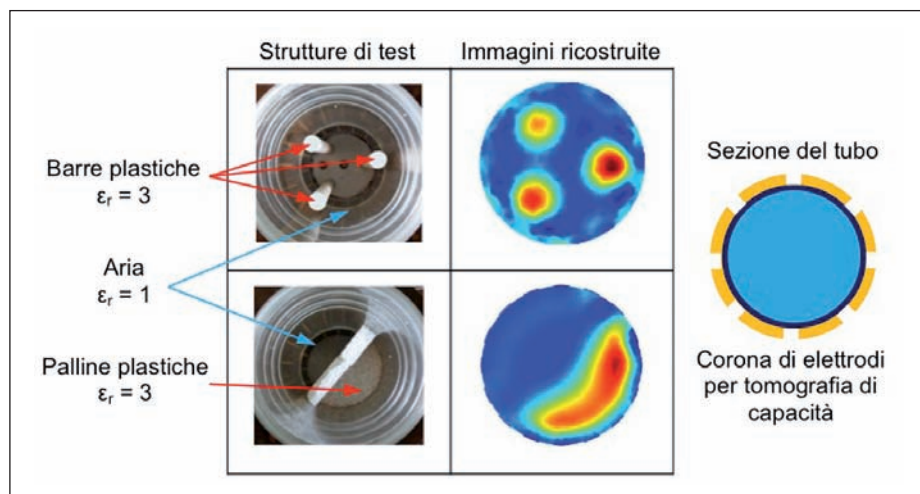



*Figura 3: esempi della caratterizzazione con strutture plastiche in aria di un sistema di tomografia capacitiva a 16 elettrodi applicati a un tubo di 10 cm di diametro.*

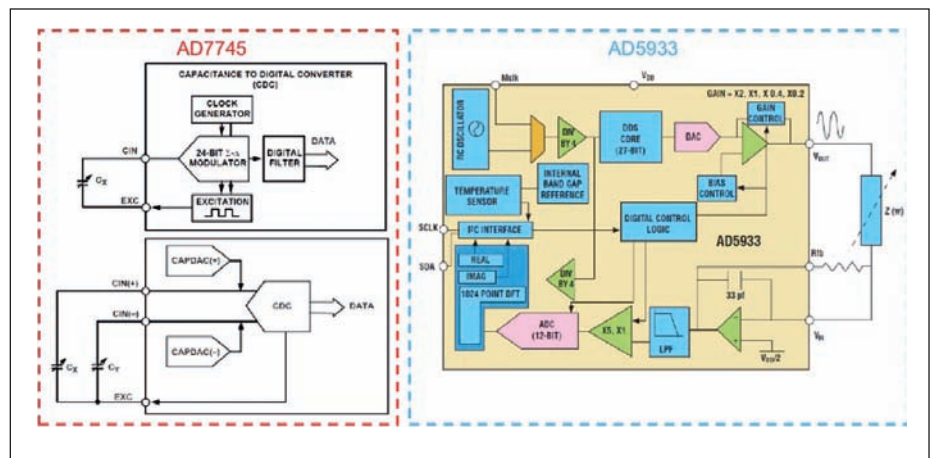



*Figura 4: schemi a blocchi funzionali di due recenti integrati di Analog Devices per la misura di capacità (AD7745) e di impedenza (AD5933).*

Infine, ragionando nel dominio del tempo, è possibile inserire la capacità in un semplice circuito RC (in cui la R è nota) che viene eccitato da uno scalino di tensione in modo da misurare la durata del transitorio di carica o scarica, ricavando dalla misura del tempo di assestamento (facilmente implementabile con il timer di un microcontrollore) il valore della costante di tempo e quindi della capacità.

Diversi circuiti integrati sofisticati con uscita digitale sono disponibili per eseguire la misura di capacità ad alta risoluzione **(figura 4)**. Ad esempio l'AD7745 contiene un oscillatore di eccitazione a onda quadra e un convertitore capacità / stream digitale basato su un convertitore analogico-digitale di tipo sigma/delta a 24 bit, consumando meno di 1 mA.

Inoltre esso permette la lettura differenziale e la compensazione attiva di eventuali offset capacitivi significativi con convertitori DAC interni denominati CAP-DAC.

Il massimo della complessità è contenuto però nell'AD5933 che racchiude un generatore di segnale sinusoidale di tipo DDS (Direct Digital Synthesis) fino a 100 kHz, uno stadio di ingresso a transimpedenza, un convertitore ADC a 12 bit e una logica digitale che esegue la trasformata di Fourier discreta (DFT) su 1024 campioni acquisiti per calcolare la parte reale (resistiva) e immaginaria (capacitiva) dell'impedenza incognita collegata Z(w), con modulo da 100 Ω a 10 MΩ.

## INTERFACCE UOMO-MACCHINA

La rapida diffusione di tablet e di dispositivi palmari con schermo tattile ha accelerato bruscamente l'interesse verso le interfacce uomo-macchina di tipo a sfioramento anche in applicazioni non consumer. Anche in questo caso l'approccio capacitivo è vincente in quanto è possibile

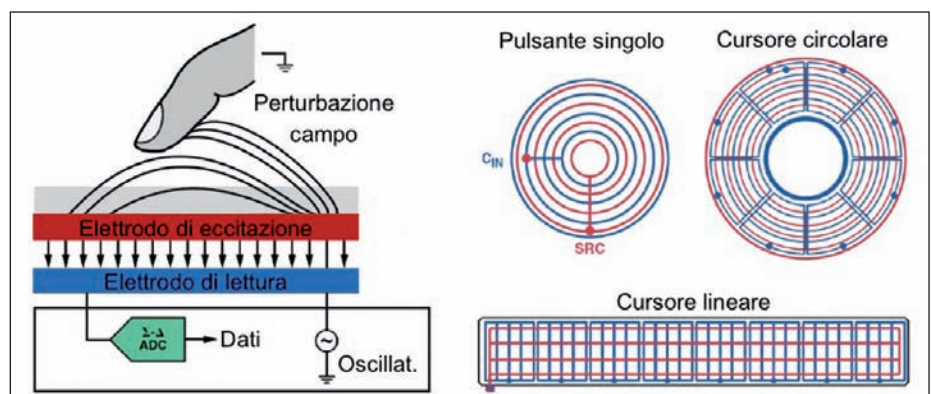



*Figura 5: principio di funzionamento delle interfacce utente capacitive e varie geometrie di pulsanti e cursori.*

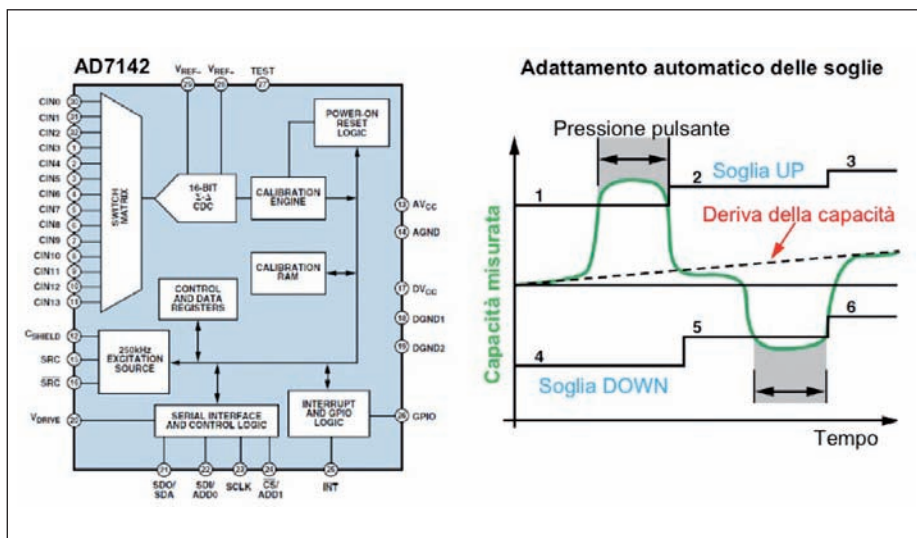



Figura 6: integrato dedicato per la gestione di 14 interruttori capacitivi con adattamento automatico delle soglie di sfioramento.

rilevare la presenza e la posizione del dito dell'utente senza la necessità di una pressione meccanica.
Si misura la capacità tra una coppia di elettrodi affiancati o sovrapposti **(figura 5)** tipicamente ricoperti da un sottile pannello isolante, su cui scivolano dolcemente le dita. Sebbene la maggior parte delle linee di forza del campo elettrico siano comprese tra i due elettrodi, alcune linee (responsabili del fringing field) fuoriescono, interessando la superficie del pannello in modo che la presenza del dito possa creare una perturbazione misurabile. Per questo motivo la capacità fissa è dell'ordine dei picoFarad mentre le sue variazioni dovute alle dita sono sui femtoFarad.

Tipicamente gli elettrodi sono fabbricati utilizzando le piste metalliche di normali PCB a doppia faccia e organizzati in matrici. Alcune possibili geometrie di interruttori e di cursori sono illustrati in **figura 5**.
Esistono anche in questo caso singoli integrati dedicati che facilitano la gestione di tali interfacce. Ad esempio l'AD7142 **(figura 6)** permette il controllo intelligente di 14 sensori tramite un multiplexer integrato assieme a un circuito di eccitazione a 250 kHz (elettrodo comune a tutti i sensori) e un convertitore sigma/delta capacitivo a 16 bit e logica di gestione con uscita su bus seriale (SPI o I2C). Inoltre esso implementa un algoritmo di calibrazione automatica delle soglie che permette di seguire la deriva del valore di capacità durante le fasi di inattività (dovute ad esempio a transitori termici) e quindi di adattare dinamicamente i valori delle soglie di scatto. ❑

**CODICE MIP 2830015**

## PER approfondire...


- www.sensortechnics.com
- S. Pratt, "Capacitance Sensors for Human Interfaces to Electronic Equipment" in "Ask the Application Engineer – 35", Analog Dialogue 40–10, 2006.
- "Introduction to capacitive sensing" Microchip, AN1101.

INFO

INVESTIAMO NEL FUTURO

a cura della redazione

Il punto vendita e centro Logistico di Vimodrone (MI).

# 20 anni DI RS COMPONENTS

*Lo scorso 14 settembre, presso la spettacolare sede dell'autodromo di Monza, la festa dei 20 anni della filiale Italiana di RS.*

La sede di RS Components a Cinisello Balsamo (MI).

*Ha preso parte all'evento lo storico fondatore Angelo Di Benedetto, che ha raccontato appassionatamente come con un team di dodici persone, contro ogni pronostico, fece partire un business nuovo per l'epoca, che raggiunse 500.000 euro di fatturato dopo soli 6 mesi di attività e il break-even point dopo soli tre anni.*

*Ermanno Maffè, attuale Responsabile Marketing Europa, ha puntualizzato i fattori e le tappe di un successo straordinario nel corso degli anni. Il web ha rappresentato un business strategico per l'azienda, è stato infatti raggiunto nel 2009 il 50% degli ordini gestiti online. Le collaborazioni con i fornitori strategici sono state anch'esse oggetto di forte focalizzazione. L'introduzione di SAP ha permesso di gestire processi condivisi e lo scambio efficace di informazioni all'interno del gruppo.*

*Didier Goguelin, VP Sales Europa ha invece posto l'accento sulle persone: il team è tutto per RS, solo grazie a questo aspetto fondamentale ha annunciato con forza un obbiettivo molto ambizioso, ovvero raddoppiare il fatturato in Italia, che rappresenta un key market per RS.*

*Maurizio Candura, Country Manager Italia, ha invece sottolineato come il portafoglio prodotti di RS è cresciuto sino all'incredibile cifra di 100.000 prodotti a catalogo cartaceo mentre il web ne conta ben 550.000. I numeri impressionanti non sono finiti: 3.000 fornitori e 350.000 ordini gestiti con un fatturato di 70 milioni di euro e 250 dipendenti fanno di RS un'azienda di prim'ordine nel panorama della distribuzione elettronica. Il raggiungimento di questi numeri si basa su un'idea semplice:*

- *catalogo gratis;*
- *spedizione entro 24 ore;*
- *spedizione gratis (all'inizio);*
- *confezioni singole;*
- *nessun minimo ordinabile;*
- *copertura nazionale;*
- *qualità e affidabilità dei prodotti.*

*L'innovazione è nel processo logistico mentre l'e-commerce è strutturato seguendo un approccio "human touch". La presenza nei media di RS è numericamente a effetto, con 4.500 pagine pubblicitarie online e offline e 5.000 PR di articoli e interviste. La soddisfazione del cliente è misurata attraverso il TRI*M-index, standard internazionale, che vede solo cinque aziende al livello di RS. Altro fattore essenziale per RS è la responsabilità sociale, con zero incidenti sul lavoro e rispetto del territorio.*

*Oltre lo staff di RS, hanno preso parte all'evento portando testimonianze di successo due fornitori strategici: Agilent e Fluke, ma anche un cliente, Edison, che ha puntualizzato come l'importanza dell'e-procurement integrato e il sistema "punch-out" ha permesso un enorme risparmio di tempo e azzeramento degli errori nell'emissione ed evasione degli ordini.*

*Sono intervenuti anche la Fondazione "La Pelucca" e il sindaco di Cinisello Balsamo, ove RS ha la sua sede, a testimoniare come l'impegno e l'attenzione di RS verso il proprio territorio abbiano rappresentato un fattore di esclusività e di eccellenza, attivando progetti per la cura degli anziani, sostegno al lavoro e per la diffusione della cultura nelle scuole medie inferiori.*

## RS COMPONENTS

*RS Components e Allied Electronics sono marchi commerciali di Electrocomponents plc, il maggiore distributore mondiale di prodotti di elettronica, manutenzione e industriali.*

*Il Gruppo è presente con sedi operative in 32 Paesi e, attraverso Internet e i cataloghi cartacei, distribuisce una gamma di oltre 550.000 prodotti a più di un milione di clienti in tutto il mondo, evadendo oltre 46.000 ordini il giorno stesso in cui essi sono ricevuti.*

*Gli articoli distribuiti, provenienti da oltre 2.500 fornitori leader, includono componenti elettronici, di automazione e controllo, test e misura, elettrici e meccanici.*

*Electrocomponents è quotata alla Borsa di Londra (London Stock Exchange), e ha chiuso lo scorso anno finanziario il 31 marzo 2012 con un fatturato di 1,27 miliardi di sterline.*

*Per maggiori informazioni: www.rs-components.com*

di GIOVANNI DI MARIA

# PRIMO APPROCCIO *al* RASPBERRY PI

**Tocchiamo con mano il Raspbery Pi e scopriamo assieme le possibilità che ci offre questo piccolo grande gioiello tecnologico dal prezzo estremamente economico.**

*Figura 1: la confezione contenente il Raspberry Pi.*

Finalmente è arrivato. Il PC più piccolo del mondo. Ma si tratta ancora di un dispositivo estremamente difficile da reperire e avere tra le mani. Da tutto il mondo vi sono milioni di ordini e di prenotazioni e l'azienda che lo produce fatica a stare al passo di questa colossale domanda.
L'articolo che vi accingete a leggere ha lo scopo di esaminare il Raspberry Pi in dettaglio. Si scoprirà innanzitutto di che cosa si tratta; prove pratiche e sperimentazioni saranno alla base del nostro lavoro. Per gli approfondimenti e la sua programmazione, nonché la realizzazione di dispositivi elettronici, dovete invece aspettare le prossime puntate di Fare Elettronica.
E adesso... andiamo ad aprire la confezione.

## UNA RAPIDA OCCHIATA

Un oggetto così piccolo ma tanto potente non può che destare tanta ammirazione. In effetti un computer completo nello spazio di pochi centimetri non è una cosa che si vede ogni giorno. Fa parte infatti della categoria dei picocomputer. La sua leggerezza e la compattezza sono le caratteristiche che saltano subito agli occhi e ci si domanda come sia stato possibile inglobare tante funzioni e tanti diversi dispositivi in così poco spazio.
Esaminandolo e ruotandolo tra le mani si scorgono le porte e i connettori che permettono il collegamento del sistema al mondo esterno. Il Raspberry Pi viene fornito "nudo", ossia non è contenuto dentro un case. Tra qualche puntata è prevista la pubblicazione di un articolo che tratterà appunto dei contenitori e della loro grande varietà e spiegherà anche alcune tecniche per la loro realizzazione in maniera semplice e veloce.

Una parte consistente del circuito è occupato dal suo tipico logo, ossia un bel lampone che è anche riproposto sul desktop del S.O, in versione colorata.
Che cosa troviamo sul Raspberry Pi? Proviamo ad elencare le componenti più importanti:

- due prese USB 2.0 espandibili con Hub;
- una porta Ethernet 10/100 (solo per il modello B);
- una uscita HDMI per il collegamento a un monitor o un TV dotato di tale ingresso;
- una porta di alimentazione a 5 volt con connettore micro USB (per alimentare il Raspberry Pi si può utilizzare anche il caricabatterie del proprio telefonino);
- un'uscita RCA video composita, da collegare a un televisore;
- un'uscita audio jack;
- un lettore SD card che permette l'alloggiamento e la lettura del sistema operativo, nonché dà la possibilità di salvare e memorizzare i propri dati, come se esso fosse un vero e proprio hard disk;
- un connettore GPIO (General Purpose Input/Output) che contiene ingressi e uscite digitali, per qualsiasi applicazione utente. Gli dedicheremo un apposito articolo per descriverlo in dettaglio.

## EFFETTUIAMO IL PRIMO COLLEGAMENTO

Bene, siamo pronti per effettuare il primo collegamento "minimo" per permetterne subito il collaudo e osservarne il funzionamento. Ovviamente abbiamo bisogno di una SD card formattata con il S.O. per il Raspberry Pi, dal momento che la confezione ne è sprovvista. In una delle prossime puntate si tratteranno i metodi per la

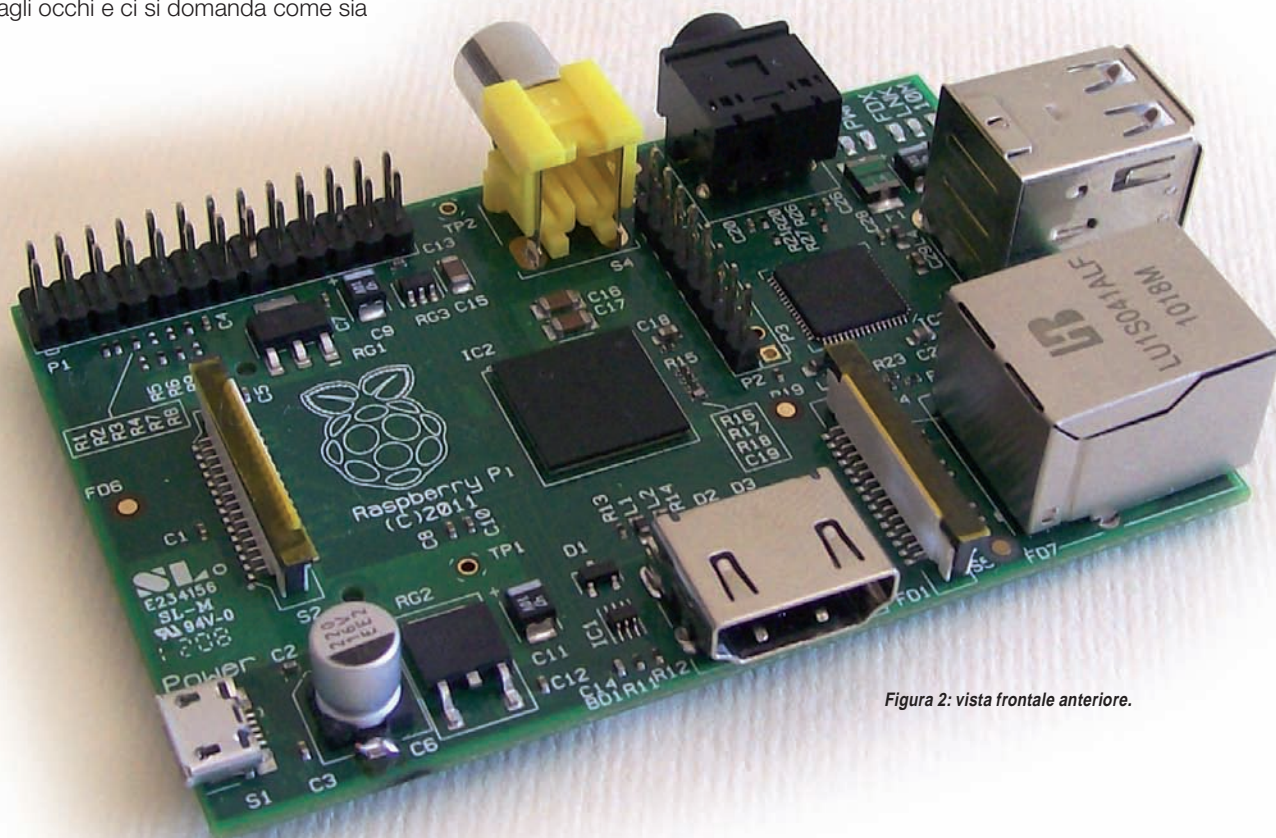



*Figura 2: vista frontale anteriore.*

preparazione di una SD, in ambiente Windows e Linux.
I componenti hardware minimi da possedere sono, ovviamente, un televisore, dotato di ingresso video RCA (anche tramite adattatore SCART), una tastiera USB e un mouse USB da collegare negli appositi connettori. È possibile utilizzare un HUB per moltiplicare le uscite USB. È richiesto, naturalmente, un alimentatore da 5,1 V con connettore micro USB. Può essere utilizzato con successo anche il caricabatterie del telefono cellulare, purché riesca a erogare almeno 700 mA di corrente. Come vedremo successivamente, il Raspberry Pi può anche essere alimentato fornendo una tensione di 5,1 Volt alla porta GPIO sul pin 2.

TABELLA 1:
CARATTERISTICHE TECNICHE DEL RASPBERRY PI MOD. B

| | |
|---|---|
| **Dimensioni** | **85,6 mm × 53,98 mm × 17 mm** |
| **Processore** | **700 MHz ARM** |
| **RAM** | **256 MB** |
| **Memoria di massa** | **scheda SD con Linux** |
| **Rete** | **Ethernet 10/100 BaseT** |
| **Video** | **Presa HDMI, Presa video RCA** |
| **USB** | **Due prese USB 2.0** |
| **Audio** | **Uscita per jack da 3,5 mm** |
| **Peso** | **45 grammi** |
| **Alimentazione** | **Micro USB** |
| **Porta I/O** | **GPIO** |

## ADESSO ACCENDIAMOLO

Una volta effettuati i collegamenti, proprio come un vero e proprio PC, possiamo dare tensione al sistema e accendere la TV. Ovviamente essa deve essere sintonizzata sul canale AV1 o AV2, dal momento che si tratta di un ingresso analogico esterno. Si vedrà immediatamente la videata che mostra il boot in corso, ossia il processo di avviamento del Raspberry Pi, con il sistema operativo Linux, su piattaforma Debian. Il tempo di caricamento dipende dalla distribuzione del sistema operativo e dalla tipologia (classe) della SD. Esso può durare da poco meno di un minuto a un paio di minuti al massimo. Ricordiamolo, si tratta di un sistema ope-

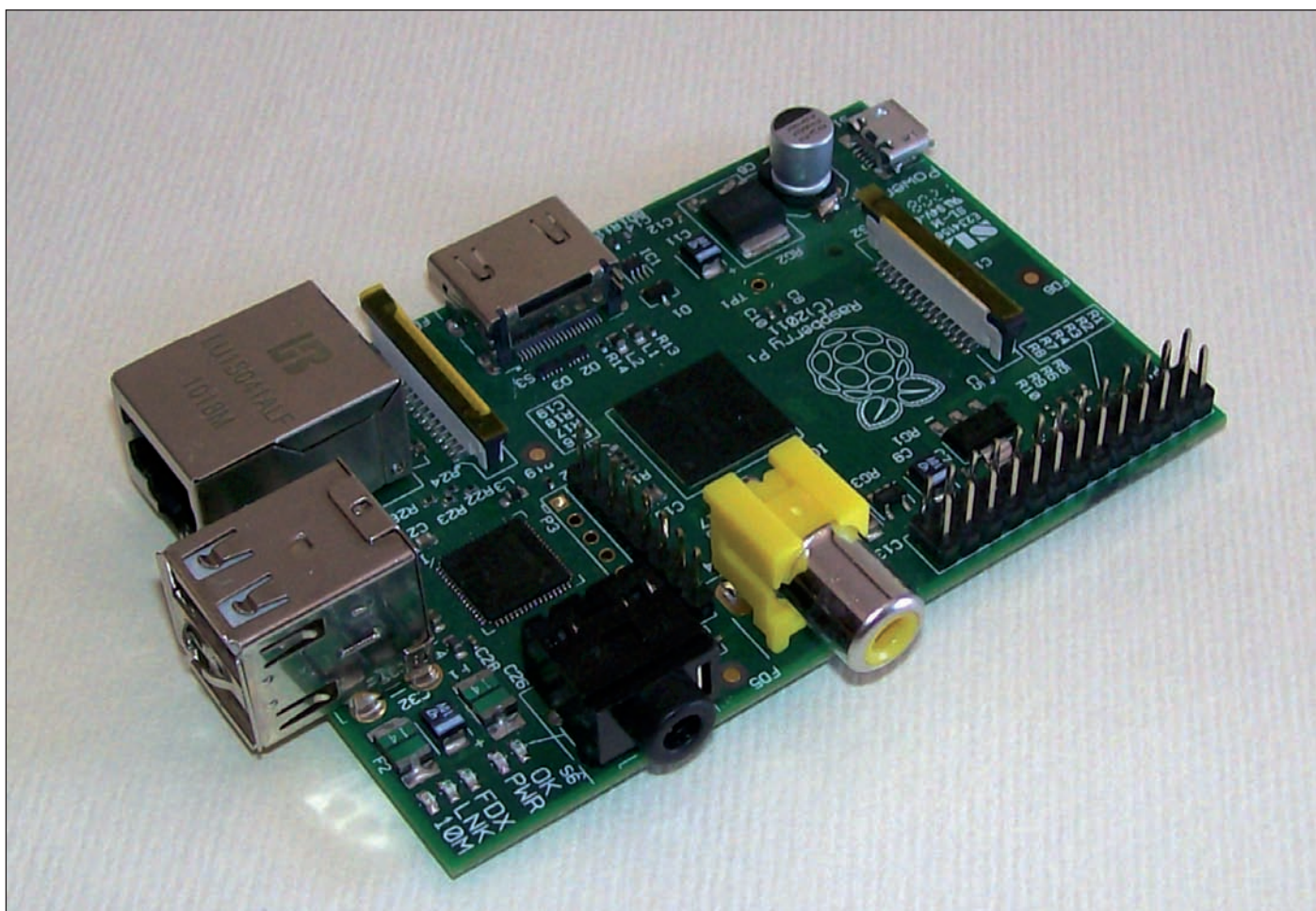



*Figura 3: vista posteriore.*

*Figura 4: vista dal basso.*

rativo molto snello e leggero, per cui la scelta di adottarlo per il Raspberry Pi è stata estremamente azzeccata. Al termine del processo di boot il sistema chiede di inserire una username e una password. Essa è differente a seconda di quale distribuzione si sta utilizzando. Al momento della scrittura del presente articolo, l'accesso avviene con le seguenti credenziali, in dipendenza della distribuzione utilizzata:

- Raspbian "wheezy": Username: **pi**; Password: **raspberry**;
- Arch Linux ARM: Username: **root** Password: **root**;
- QtonPi: Username: **root** Password: **rootme**.

Dopo qualche secondo, sul video appare il classico "prompt" di sistema che attende i comandi da parte dell'utente. Se si entra come utente "root" occorre prestare la massima attenzione, poiché con un comando errato si potrebbe inavvertitamente eliminare il file system.
L'aspetto interessante: almeno per la distribuzione Raspbian "wheezy" è previsto anche l'ambiente X, per cui basta lanciare il comando startx con la tastiera per avviare e iniziare la sessione grafica per eccellenza, con l'utilizzo del mouse (LXDE).

La dotazione del software non è certo ricchissima. Mentre dal lato "console" sono presenti pressoché tutti i comandi di sistema, in ambiente X i programmi sono limitati. Ma questo dipende ovviamente dalla distribuzione adottata. Nulla vieta infatti di realizzare da sé la propria distribuzione preferita.
In alcune distribuzioni è presente il gioco Quake 3, perfettamente giocabile e con una resa stupefacente, caratterizzato da una fluidità di movimento senza eguali.
Tra le applicazioni presenti in ambiente LXDE troviamo il file manager, un comodo editor testuale, un visualizzatore di immagini, alcune finestre di console di comando, alcune applicazioni educative e altro. Per Internet vi sono ben tre browser: Dillo, Midori e NetSurf Web.

Per i programmatori, e questo ci interessa da vicino, c'è il compilatore universale "gcc" per compilare i propri programmi in linguaggio C e, soprattutto, è presente il potente linguaggio Python. Nel corso delle nostre future sperimentazioni, entrambi i linguaggi saranno utilizzati in modo molto approfondito.

## TASTIERA ITALIANA

Quasi sicuramente, al momento del boot, il sistema, per default, si troverà con la configurazione della tastiera inglese per cui molti tasti, specialmente quelli che si trovano sopra i numeri, non corrispondono con i loro reali simboli. Dal prompt è quindi opportuno abilitare la tastiera ita-

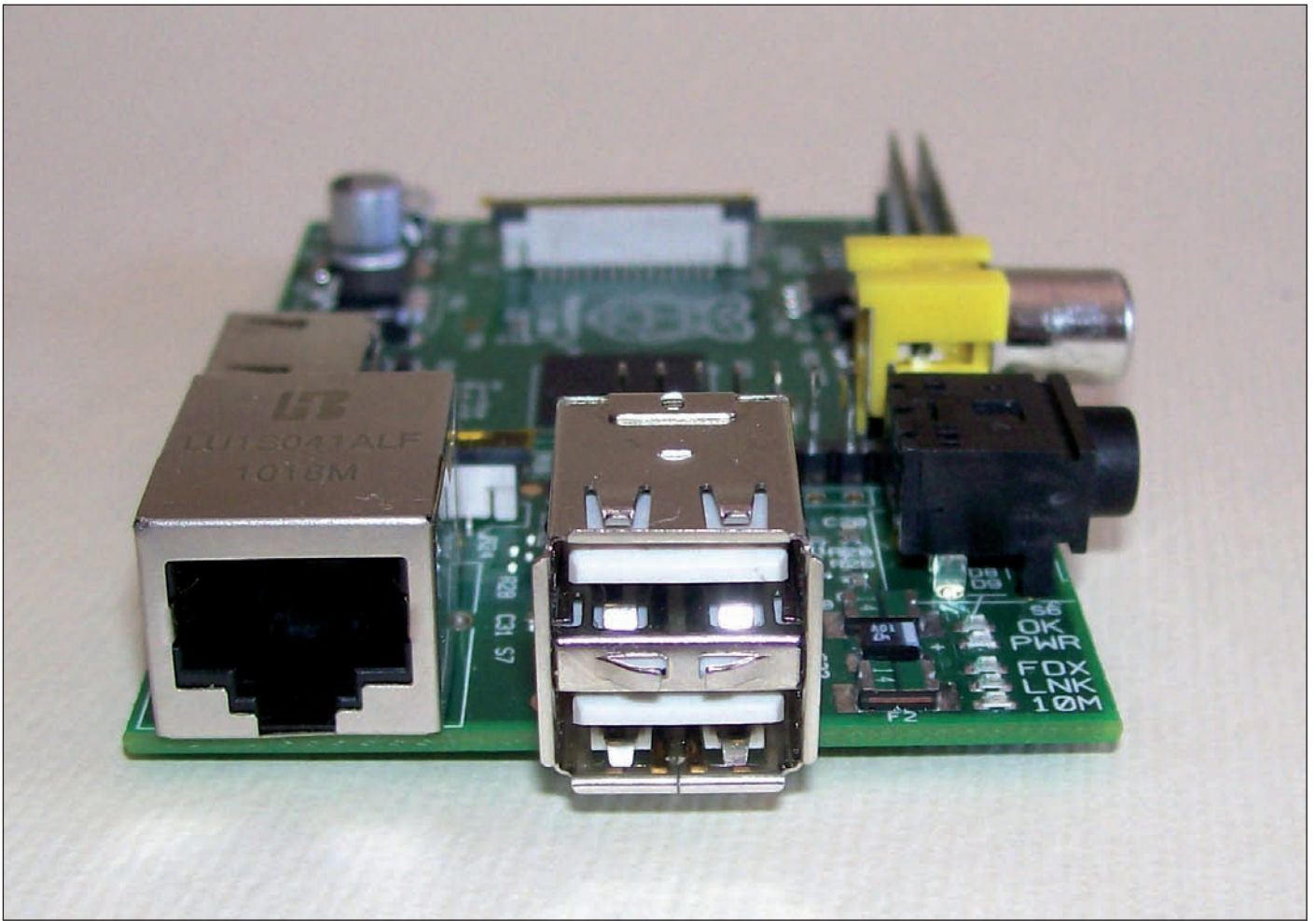



Figura 5: vista laterale.

liana. Con i privilegi root o, come utente tramite il comando “sudo”, si deve impartire il seguente comando:

```
sudo dpkg-reconfigure keyboard-configuration
```

Si deve rispondere alle varie domande di configurazione che vengono proposte dal programma. Le modifiche saranno disponibili al successivo riavvio.

## INTERNET

Il Raspberry Pi, grazie all’interfaccia di rete Ethernet 10/100, permette il collegamento e la navigazione su Internet. Essa risulta molto veloce e non si percepiscono affatto i “limiti” imposti dalla CPU ARM di 700 Mhz. Con i suoi tre browser presenti, tutte le esigenze di navigazione e di ricerca dei siti sono soddisfatte. Nel prossimo futuro è prevista l’uscita di Chrome per Raspberry Pi.

Figura 6: collegamento minimo per la prova di funzionamento.

### PERCHÉ IL RASPBERRY PI È DIVERSO DA UN PC

Il Raspberry Pi, nonostante sia un computer a tutti gli effetti, dispone di alcune caratteristiche che lo rendono estremamente interessante, specialmente nel campo dell'elettronica sperimentale. Innanzitutto i consumi sono estremamente bassi, pertanto il suo utilizzo non grava minimamente sulla bolletta della luce. Inoltre ha una porta di comunicazione (GPIO) facilmente programmabile dal software, con qualsiasi linguaggio di programmazione.
Questo permette di collegare tante periferiche alle sue uscite (ovviamente opportunamente interfacciate) in modo da realizzare un sistema completo e utile.

### CAMPI DI APPLICAZIONE

I campi di applicazioni del Raspberry Pi sono infiniti. La lista che segue elenca solo alcune delle numerosissime possibilità pratiche. A grandi linee, i settori di competenza sono presentati nell'elenco sottostante, ma certamente l'eccezionalità del prodotto copre un numero maggiore di settori applicativi. Solo la fantasia infatti

```
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting system message bus: dbus.
Starting dphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
My IP address is 10.0.2.15

Debian GNU/Linux wheezy/sid raspberrypi tty1

raspberrypi login: root
Last login: Thu Aug 16 20:36:20 UTC 2012 on tty1
Linux raspberrypi 3.1.9+ #2 Mon Apr 16 04:53:15 EST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

root@raspberrypi:~#
```

*Figura 7: boot completato, login effettuato e prompt dei comandi.*

```
root@raspberrypi:~# pwd
/root
root@raspberrypi:~# who am i
root     tty1         2012-08-17 10:42
root@raspberrypi:~# ls -l
total 4
drwxr-xr-x 2 root root 4096 Jul 31 19:51 Desktop
root@raspberrypi:~#
```

*Figura 8: alcuni comandi di Linux.*

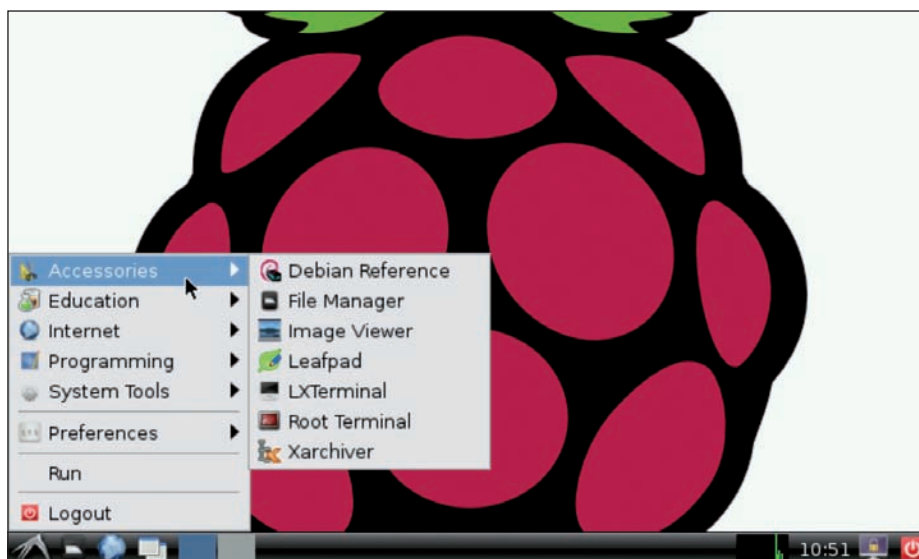



Figura 9: LXDE in azione.

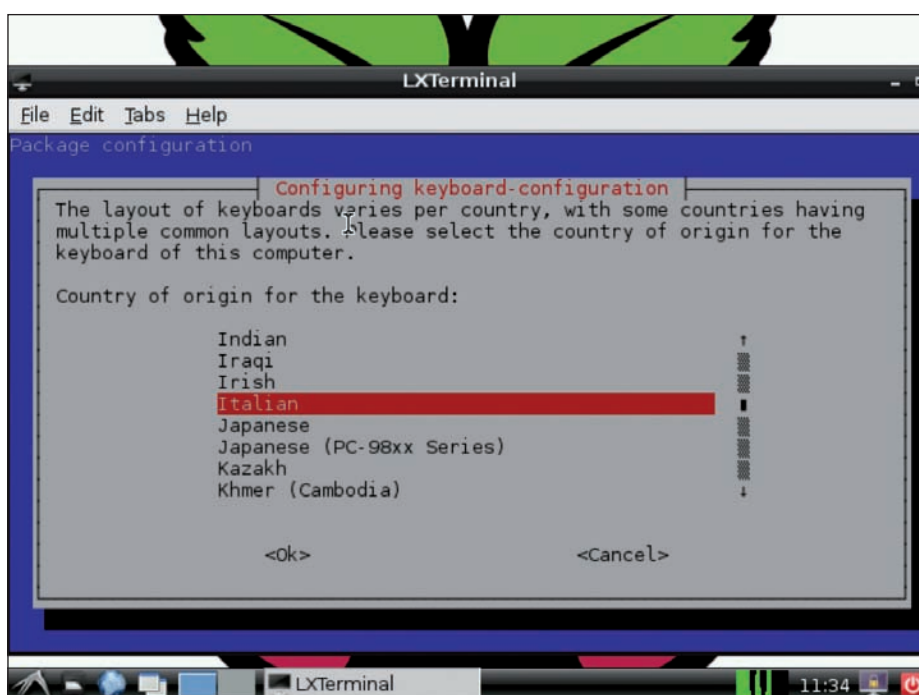



Figura 10: configurazione tastiera italiana.

Figura 11: Scratch, un divertente programma educativo.

Figura 12: navigazione con Raspberry Pi sul sito www.farelettronica.com.

può far navigare il progettista negli infiniti meandri delle possibilità offerte dal sistema. Alcuni settori di applicazioni sono i seguenti:

- robotica;
- domotica;
- didattica dell'informatica e della programmazione;
- divertimento e videogiochi a basso costo;
- controllo remoto;
- controllo di sistemi;
- sostituzione dell'uomo in luoghi pericolosi;
- monitoraggio zone impervie (anche ad alte quote);
- Internet radio a basso costo;
- applicazioni multimediali.

## LE APPLICAZIONI PIÙ BIZZARRE TROVATE IN RETE

Il Raspberry Pi è utilizzato per una varietà notevole di applicazioni, alcune davvero insolite. Ecco qualche idea, trovata su Internet, che potrebbe fornire al lettore alcuni spunti di realizzazione:

- monitoraggio di meteoriti;
- controllo ambiente su pallone aerostatico;
- Raspberry Pi in orbita;
- emulatore di vecchi giochi Nintendo;
- macchina erogatrice di caffè.

## ULTIME CONSIDERAZIONI

Per descrivere il Raspberry Pi occorrerebbero molte pagine. Forniamo qualche ulteriore commenti e considerazione aggiuntiva su di esso. La prima osservazione da fare è quella della indipendenza del Raspberry Pi dalla TV. Infatti, una volta programmato a dovere, esso può funzionare senza televisione e senza mouse. Ciò torna molto utile in caso di realizzazione di centraline o prototipi da dislocarsi in siti critici o comunque di difficile collocamento.

Il Raspberry Pi può costituire inoltre un ottimo regalo da fare ai bambini, in quanto unisce l'utile al dilettevole. La possibilità infatti di imparare l'informatica e la programmazione dei computer assieme all'utilizzo ludico sono potenzialità senza paragoni. È possibile, ad esempio, emulare i vecchi home computer e le vetuste console di videogame, come il Commodore 64 o il Nintendo. Inoltre il basso consumo

Figura 13: un bimbo gioca con Quake 3.

Figura 14: una videata del gioco Quake 3.

non incide minimamente sull'importo finale della bolletta dell'energia elettrica, e si può tenere acceso tranquillamente 24 ore di seguito. Tra l'altro la temperatura del dispositivo non raggiunge livelli di guardia. E consigliabile l'utilizzo di una piccola ventola solo se si prevede la dislocazione del Raspberry Pi all'interno di un piccolo contenitore o case.
In ambiente di ricerca scientifica si può prevedere di allestire un "array" di Raspberry Pi che, sebbene la limitata velocità di clock, potrebbe essere dedicato a elaborazioni matematiche particolarmente pesanti. L'unione fa la forza e l'organizzazione di – ad esempio – un migliaio di unità potrebbe fornire una buona potenza, paragonata a quella erogata da circa duecento PC "tradizionali" con un costo pari a un terzo.

## CONCLUSIONI

Il Raspberry Pi è sicuramente utilizzabile per gli usi più impensati. La sua affidabilità è massima e, durante i nostri test, non si è mai "bloccato" né ha dato alcun segno di cedimento. Il processore AMR a 700 Mhz è utilizzato egregiamente dal sistema operativo. Le nostre prove sul supporto multimediale sono state molto soddisfacenti e la navigazione su Internet non ha creato problemi di sorta.
Inoltre, i dispositivi elettrici ed elettronici sperimentali da noi realizzati, pilotabili dal sistema, hanno risposto perfettamente alle prove e non hanno mai dato segni di instabilità. Tali applicazioni sono state realizzate a scopo sperimentale nei tre linguaggio di programmazione da noi adottati, ossia il linguaggio C, il Python e l'Harbour.
In conclusione il Raspberry Pi avrà sicuramente un futuro brillante in qualsiasi campo applicativo e, soprattutto per il suo esiguo costo, sarà utilizzato in ogni settore. Consigliamo dunque a tutti di entrare in possesso di questo meraviglioso gioiello, che potrà dare non poche soddisfazioni.
Alle prossime puntate per nuovi e affascinanti esperimenti con il Raspberry Pi ❑

**CODICE MIP 2830019**

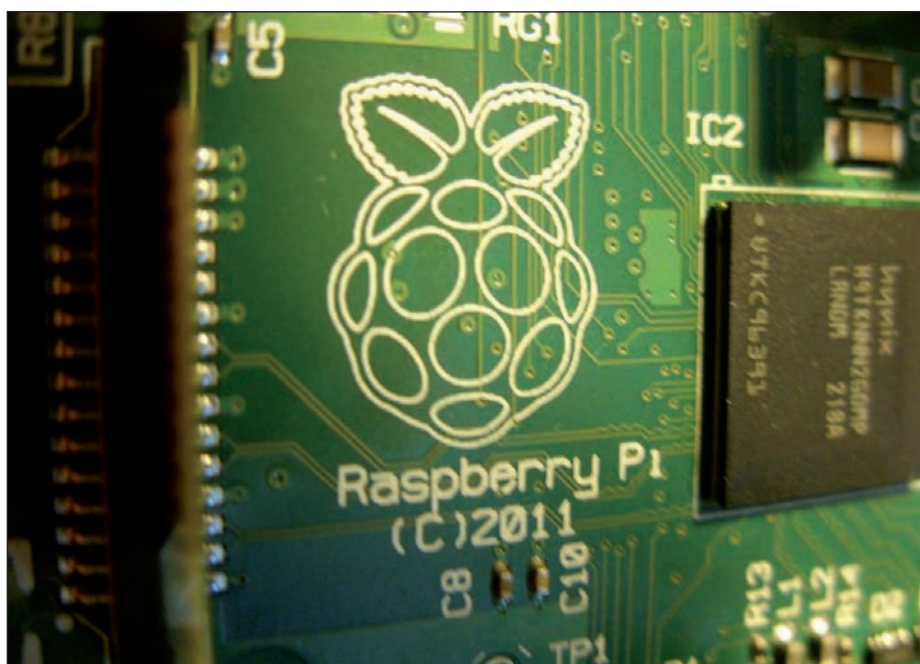



Figura 15: il lampone del Raspberry Pi.

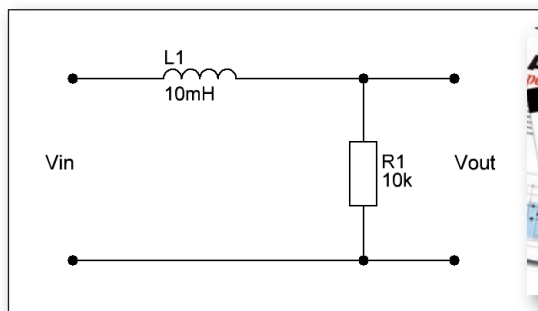
# elettro quiz 329

# rispondi & vinci

*Le risposte ai quiz "facile" e "difficile" vanno inviate esclusivamente compilando il modulo su www.farelettronica.com/eq specificando la parola chiave* ***"Farad"****.*

*Le risposte e i vincitori (previa autorizzazione) sono pubblicati alla pagina www.farelettronica.com/eq a partire dal 15 del mese successivo alla pubblicazione sulla rivista.*

*A tutti i partecipanti verrà assegnato un buono sconto del 10% (validità 3 mesi dalla data di assegnazione) utilizzabile per un prossimo acquisto su www.ieshop.it*

L1
10mH
Vin
R1
10k
Vout

abbonamento al club di Fare Elettronica

## facile

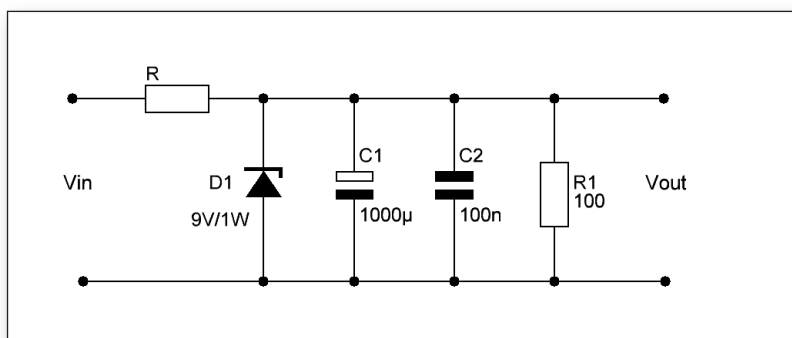
*Dato il circuito di figura, quali delle seguenti affermazioni sono vere?*

***a)*** *Il circuito è un filtro passa basso*

***b)*** *La componente continua di Vin viene bloccata*

***c)*** *Un segnale di frequenza 100 Hz passa inalterato*

***d)*** *Eventuali segnali Wi-Fi captati dall'ingresso vengono bloccati.*

R
Vin
D1
9V/1W
C1
1000µ
C2
100n
R1
100
Vout

## difficile

*Nello schema dell'alimentatore stabilizzato a diodo zener rappresentato in figura, dimensionare la resistenza R in modo che lo zener, da considerarsi ideale, dissipi 1W di potenza.*

Abbonamento a Fare Elettronica

di GIOVANNI DI MARIA

# PWM avanzato (parte prima) CON IL PIC

**In questa prima parte approfondiamo le conoscenze e le tecniche legate all'utilizzo del segnale PWM di tipo software generato dai microcontrollori PIC. Nella prossima puntata tratteremo i metodi di generazione di tipo hardware.**

## IL PWM

Un articolo sul PWM è già stato pubblicato, in precedenza, durante il corso sulla programmazione dei PIC con il mikroBasic. Non staremo pertanto a ripetere gli stessi concetti base, per i quali il lettore è invitato a rileggere i relativi articoli. Ricordiamo solamente che un segnale PWM è un'onda quadra, ovvero un treno di impulsi, di cui si può variare a piacimento il duty cycle. Dal momento che tale sequenza è estremamente veloce, il carico ad esso applicato "legge" il suo effetto come un segnale di potenza variabile, in dipendenza della larghezza dell'impulso attivo.

## PWM E PIC

Un segnale PWM, oltre che essere trattato egregiamente con circuiti e integrati dedicati, può essere generato e gestito dai microcontrollori PIC. Approfondiremo in questa sede proprio tale problematica. Tramite il PIC, Il PWM può essere generato in diverse maniere. Le più importanti prevedono le seguenti modalità:

- generazione di un treno di impulsi tramite firmware scritto appositamente. Gli impulsi sono emessi da una porta di uscita della MCU. Tale metodo è prettamente di tipo software;
- generazione del PWM attraverso moduli interni ai microcontrollori, senza la ne-



*Figura 1:effetti teorici di un segnale PWM di 100 Hz al 10%, 50% e 90%.*

*Figura 2: parametri temporali di un segnale PWM.*

cessità di controllarne continuamente il flusso software. Tale metodo è di tipo hardware ma implementabile solo sui modelli di PIC più performanti.

#### Che cosa possiamo fare con il PWM?

Una delle tecniche più interessanti del PWM è poter pilotare i transistor di potenza o i PowerMOS. In questa maniera, anche i carichi molto robusti possono essere comandati con una potenza variabile, senza utilizzare alcun elemento resistivo di limitazione. La dissipazione è pertanto minima e gli elementi attivi restano freddi. Inoltre lo spreco di potenza è tanto più limitato quanto più ridotto è il valore della RdsOn di tali componenti. Con i microcontrollori si è aperta anche una nuova strada per il progettista. Con essi, seguendo la tecnica del PWM, si possono realizzare prototipi altrimenti impossibili da attuare. Si pensi, ad esempio, ai giochi di luce natalizi in cui le lampadine (o diodi LED) si illuminano progressivamente con un'intensità variabile, dando l'immagine di una luce graduale molto piacevole. In tali circostanze la variazione di potenza è fornita, appunto, da una variazione del duty cycle del segnale PWM. Questo si traduce in una diversificazione di tensione o potenza fornita al carico. In taluni casi, infatti, PWM è sinonimo di DAC, con il quale il microcontrollore può generare una tensione analogica variabile, partendo da una sorgente iniziale digitale, capace di fornire solo due diverse tensioni di alimentazione: 5 V e 0 V, rispetto a massa.

#### Risoluzione del PWM in pratica

Con l'uso opportunamente calibrato di un segnale PWM, possono essere generate praticamente tante diverse tensioni analogiche, in dipendenza anche dalla risoluzione del PWM stesso.
Con un esempio chiariamo il concetto di

*Figura 3: schema elettrico per l'accensione progressiva di un LED (PWM software).*

Figura 4: le varie larghezze degli impulsi generati.

risoluzione del PWM. Immaginiamo che il nostro microcontrollore sia in grado di generare un'onda quadra, secondo i seguenti diversi schemi di temporizzazione:

- ON=0%, OFF=100%, corrispondenti a 0 volt;
- ON=10%, OFF=90%, corrispondenti a 0,5 volt;
- ON=20%, OFF=80%, corrispondenti a 1 volt;
- ON=30%, OFF=70%, corrispondenti a 1,5 volt;
- ON=40%, OFF=60%, corrispondenti a 2 volt;
- ON=50%, OFF=50%, corrispondenti a 2,5 volt;
- ON=60%, OFF=40%, corrispondenti a 3 volt;
- ON=70%, OFF=30%, corrispondenti a 3,5 volt;
- ON=80%, OFF=20%, corrispondenti a 4 volt;
- ON=90%, OFF=10%, corrispondenti a 4,5 volt;
- ON=100%, OFF=0%, corrispondenti a 5 volt.

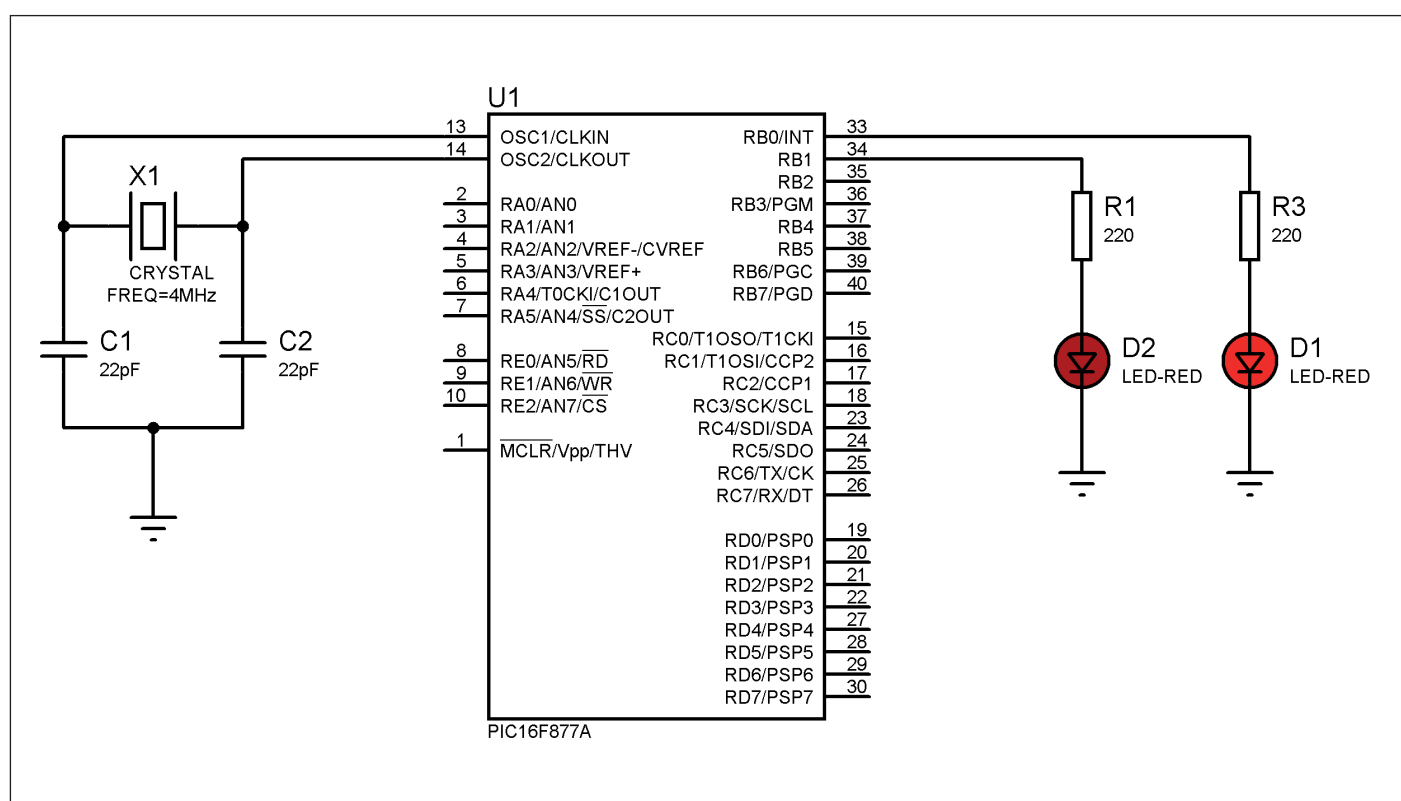



Figura 5: schema elettrico per due PWM software diversificati.

L'equivalente tensione analogica continua generata viene ricavata dalla seguente equazione:

VccMax : 100 = x : %DutyCycle

Al carico giunge un valore di potenza pari al valore medio dell'onda quadra prodotta. Nell'esempio sopra riportato, si possono generare pochi scaglioni di duty cycle, limitando il numero di combinazioni possibili.
Come si vede, dunque, il minimo salto di tensione è di 0,5 volt che corrisponde alla risoluzione. Una risoluzione di 0,5 volt è insufficiente nella maggior parte dei casi, poiché impedisce l'erogazione di tensioni intermedie, come ad esempio quella di 2,83 volt.
Una prima tecnica per migliorare i risultati è quella di poter spaziare il più possibile la scelta del duty cycle e scegliere qualunque valore (ad esempio: ON=17%, OFF=83% oppure ON=51%, OFF=49%). Per far questo è necessario mantenere la durata dell'intero periodo di una transizione completa a livelli non troppo bassi, nel qual caso sarebbe la MCU ad avere poi problemi.
Un duty cycle del 100% equivale a un segnale continuo della tensione pari a VCC, mentre un duty cycle dello 0% equivale, ovviamente, a un potenziale di massa.

## CENNI SUL PWM SOFTWARE E SUE PROBLEMATICHE

Daremo adesso alcuni spunti di utilizzo per quanto riguarda il pilotaggio del PWM via software, dedicandoci maggiormente a quello hardware, più preciso e indipendente.
La generazione di un segnale PWM via software è abbastanza semplice. Occorre eseguire qualche calcolo matematico-temporale e tradurre le pause e i ritardi in altrettante istruzioni di pilotaggio delle porte in uscita.
L'aspetto negativo di questa tecnica sta nel fatto che essa, se non perfettamente funzionante e ben analizzata, tiene troppo "occupato" il PIC. Nonostante tutto si può realizzare un buon sistema di generazione segnale PWM.
In linea di massima, con tale metodo, il segnale modulato viene generato accendendo e spegnendo opportunamente una porta d'uscita. Tale cadenza deve essere molto veloce e il tempo di accensione determina il duty cycle finale.

85%
15%

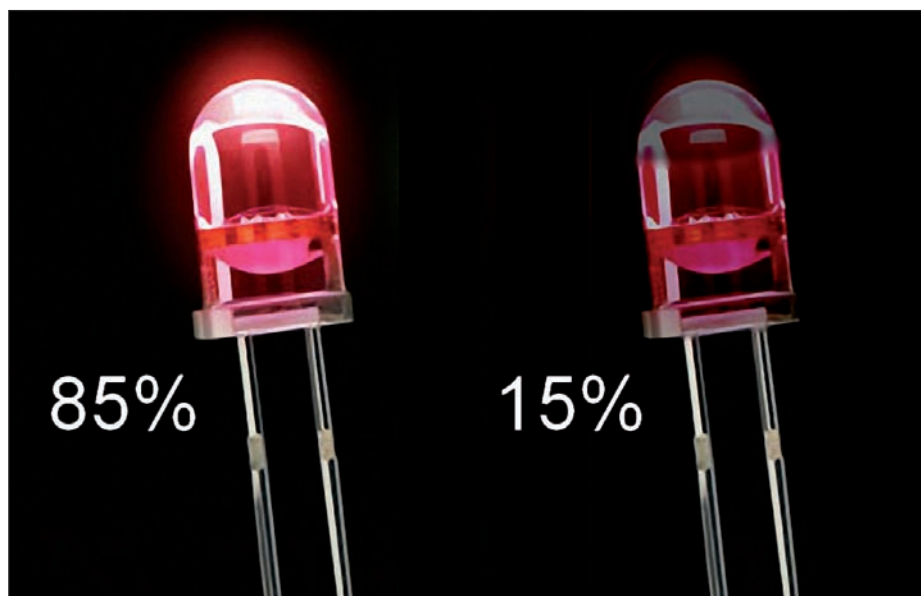



*Figura 6: due LED alimentati con PWM dell'85% e del 15%.*

### Accensione progressiva di un diodo LED

Proviamo subito la generazione del segnale PWM creando un semplice circuito che illumina progressivamente un diodo LED. Inizialmente il diodo è spento, ma pian poco alla volta la sua luminosità aumenta gradualmente, dando l'impressione che la sua fonte di energia sia, appunto progressiva. In effetti e così, anche se a cambiare non è la tensione ma la

85%
15%

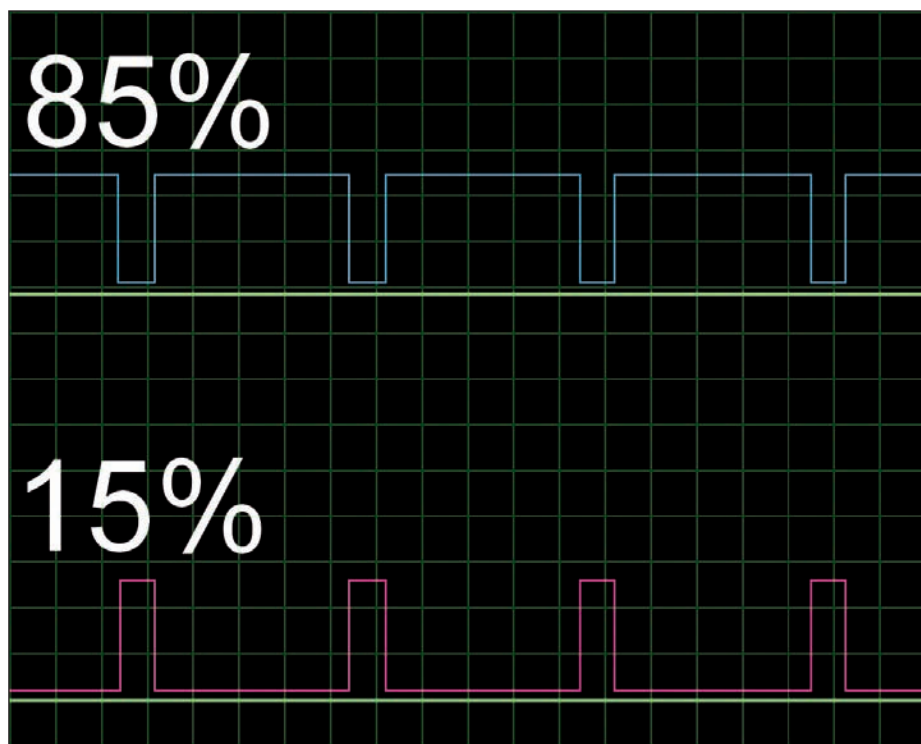



*Figura 7: i due segnali PWM all'85% e al 15% applicati ai diodi LED.*

LISTATO 1

```
rem **************************
rem *       LISTATO 1         *
rem *      PWM Software       *
rem * Accensione Progressiva *
rem *  by Giovanni Di Maria   *
rem **************************
program pwm
dim k,t_on,t_off as word
dim v as byte
main:
   trisb.0=0
   while true
         for k=0 to 10
             for v=1 to 10
                 t_on=k
                 t_off=10-k
                 portb.0=1
                 vdelay_ms(t_on)
                 portb.0=0
                 vdelay_ms(t_off)
             next v
         next k
   wend
end.
```

quantità di impulsi attivi che colpiscono il carico. È come un piccolo bombardamento che aumenta sempre più.

#### Schema elettrico per accensione progressiva di un diodo LED

Dal momento che l'implementazione del metodo è squisitamente di tipo software, qualunque tipologia di MCU va senz'altro bene. Non è infatti hardware-dipendente. Si tratta di un semplice schema che utilizza un microcontrollore 16F877. La porta RB0, configurata come linea di uscita, è collegata a un diodo LED preceduto dalla relativa resistenza di limitazione. Il clock della MCU è fornito dal quarzo assieme ai due condensatori ceramici da 22 pF.

#### Listato

Vediamo adesso come sia possibile incrementare gradatamente e lentamente la luminosità del diodo LED. Con la tecnica del PWM, tale operazione risulta estremamente semplice.
Si tratta, in pratica, di "bombardare" il LED con impulsi variabili. L'elemento cangiante è rappresentato dalla larghezza attiva dell'impulso, che aumenta progressivamente. Se gli impulsi sono di frequenza sufficientemente alta, non saranno affatto notati, per via dell'effetto persistenza della retina. Ricordiamo che per impulso attivo si intende quello con potenziale logico alto, mentre passivo è quello con potenziale logico basso, ovvero a massa (nel caso di logica positiva).
Il **listato 1** è molto semplice. Il ciclo "for" governato dalla variabile "k" imposta la larghezza degli impulsi. Più precisamente, le due variabili t_on e t_off sono calcolate per contenere due valori opposti: quando una è piccola, l'altro è grande (ad esempio: t_on=2 e t_off=8). Queste due variabili regolano il tempo nel quale il diodo LED resta acceso (t_on) e spento (t_off) creando il PWM.
Il ciclo interno, rappresentato dalla variabile "v" ripete invece più volte tale segnale PWM, consentendo un effetto più duraturo.
In definitiva il listato crea ben 11 stati differenti di larghezza di impulso. In effetti essi sono di numero limitato e l'aumento della luminosità del LED è caratterizzato da una sorta di andamento "a scatti" anziché essere perfettamente fluido. In seguito vedremo come risolvere tale problema.

#### Se un evento blocca il PWM

Il listato appena visto funziona bene, ma che cosa succederebbe se, ad esempio, durante il gioco di luce, si volesse rilevare la pressione di un tasto?

LISTATO 2

```
rem **************************
rem *       LISTATO 2         *
rem *      PWM Software       *
rem *  Due Led al 85% e 15%   *
rem *  by Giovanni Di Maria   *
rem **************************
program pwm
main:
   trisb.0=0
   trisb.1=0
   while true
         portb.0=1
         portb.1=0
         delay_us(850)
         portb.0=0
         portb.1=1
         delay_us(150)
   wend
end.
```

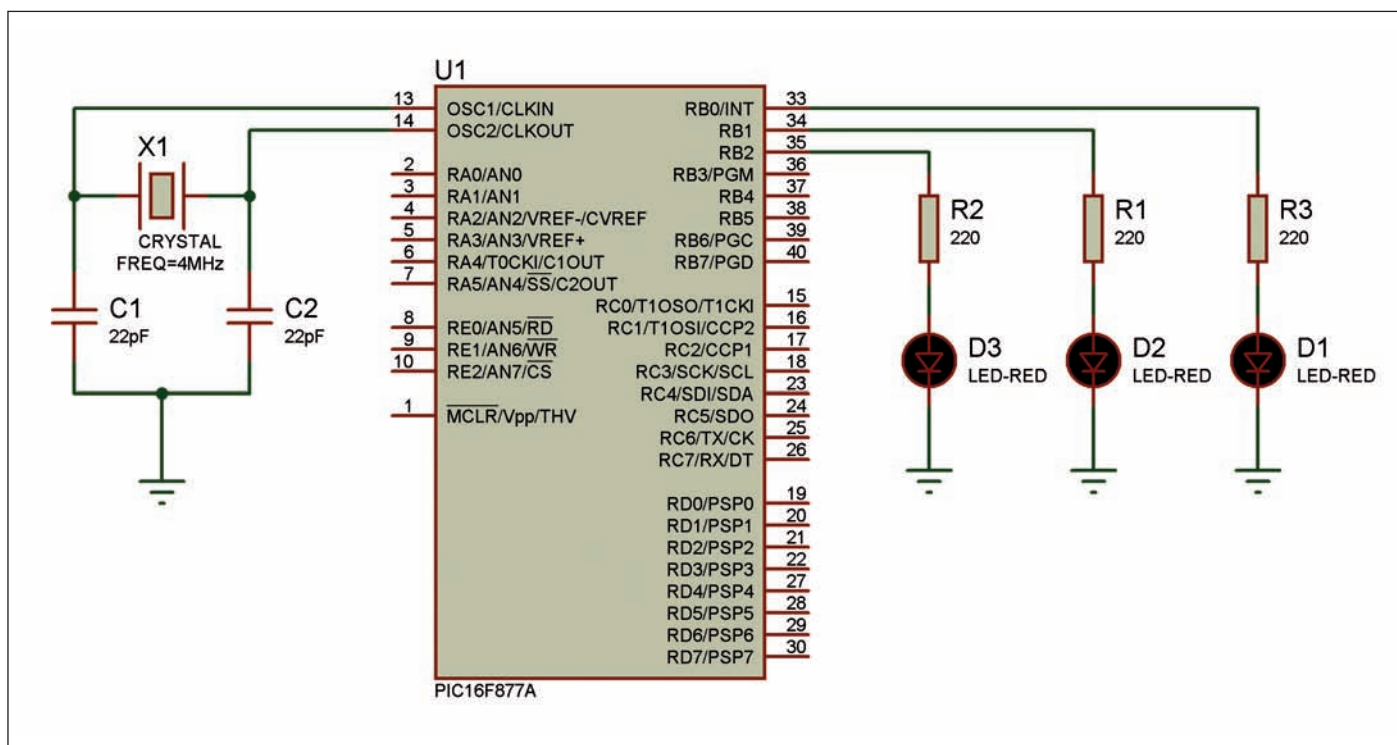



*Figura 8: schema elettrico per tre PWM software indipendenti.*

Gli effetti, se non sapientemente programmati, sarebbero a dir poco disastrosi. La rilevazione di un tasto, durante la generazione del PWM, interferirebbe con esso, rallentandolo o addirittura bloccandolo completamente. Non ci troviamo, infatti, davanti a un sistema multitasking. La soluzione al problema è estremamente difficoltosa. Si dovrebbe, in qualche modo, generare un segnale PWM anche durante la rilevazione del tasto. Oppure utilizzare le routine di interrupt, già viste e studiate nelle precedenti puntate di Fare Elettronica. Con la generazione PWM di tipo hardware questa problematica viene brillantemente risolta, in modo facile e indolore.

## DUE O TRE SEGNALI PWM SIMULTANEI

Parlando sempre di PWM di tipo software, studiamo adesso un'altra tecnica, quella di produrre due o più segnali PWM attraverso altrettante porte di uscita. La soluzione è più complicata, dal momento che, in una stessa routine iterativa, si devono pilotare tante porte, ognuna con diverso duty cycle e indipendente. Si consiglia di utilizzare la medesima frequenza di segnale per tutte le porte.

### Due PWM con duty cycle complementari

La massima semplicità si ha quando occorre produrre due segnali PWM opposti e complementari. Ad esempio, se la porta 1 genera un duty cycle del 20%, la porta 2 deve produrre un duty cycle del 80%. Oppure ancora la porta 1 del 40% e la porta 2 del 60%. Le percentuali di lavoro sono complementari poiché opposte e la loro somma equivale a 100%.
In questi casi è sufficiente generare un solo segnale PWM e dimensionarne il secondo per differenza algebrica.

Questo esempio prevede la generazione di un segnale PWM (via software) sulla porta RB0 con duty cycle del 85% e sulla porta RB1 del 15%. L'effetto finale è quello di illuminare il primo diodo LED con potenza molto maggiore rispetto al secondo, ottenendo una diversa illuminazione degli stessi.
Ma la cosa interessante è un'altra: se misuriamo la tensione ai capi delle porte RB0 e RB1, scollegando ovviamente i diodi LED, leggeremo rispettivamente i valori di 4,25 V e 0,75 V (calcolati su un

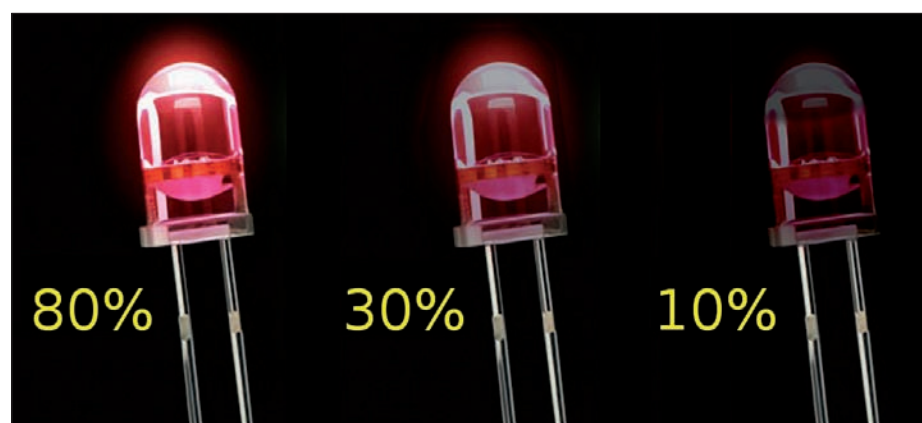
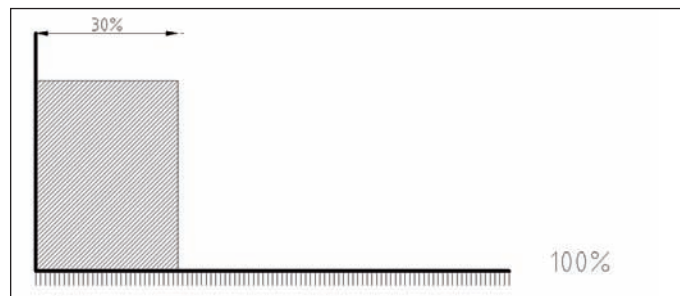
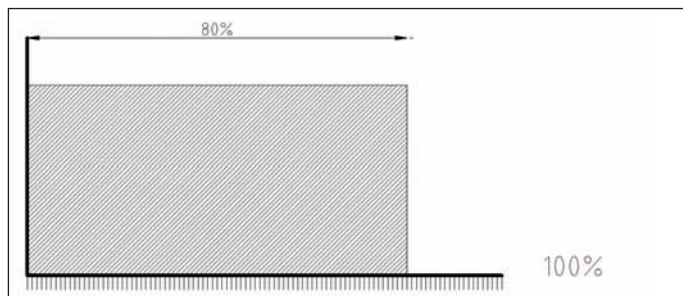



*Figura 9: tre LED alimentati con PWM software indipendente dell'80%, 30% e 10%.*

potenziale VCC di 5 V). L'oscilloscopio, collegato a queste due porte, ci informerà invece che i segnali generati sono onde quadre con ampiezza diversa.
Esaminiamo il **listato 2**. Il programma "ruota" attorno al ciclo infinito while/wend. Ogni giro completo dura circa 1.000 microsecondi (850 uS + 150 uS), equivalenti a 1 millisecondo, più una manciata di microsecondi per l'accensione e lo spegnimento delle porte RB0 e RB1. (Per la precisione diciamo che ogni ciclo dura 1.007 microsecondi). La frequenza è dunque molto alta per poter percepire l'effetto "lampeggio" dei LED. Tale ciclo infinito è composto da due segmenti: le prime tre righe (portb.0=1, portb.1=0 e delay_us(850)) accendono il primo LED e spengono il secondo LED.
Tale stato dura per ben 850 microsecondi, ossia l'85% dell'intero periodo dell'impulso. Le successive tre righe (portb.0=0, portb.1=1, delay_us(150)) spengono i primo LED e accendono il secondo. Tale stato dura "solamente" 150 microsecondi, equivalenti al 15% del ciclo completo. La sequenza, ripetuta all'infinito, causa una luminosità alta per il LED collegato sulla porta RB0 e una scarsa lucentezza per quello collegato su RB1.

### LISTATO 3

```
rem **************************
rem *       LISTATO 3        *
rem *     PWM Software       *
rem * Tre Led: 80%, 30%, 10% *
rem *  by Giovanni Di Maria  *
rem **************************
program pwm
dim p1,p2,p3 as byte
main:
     rem ——3 porte in Output——
     trisb.0=0
     trisb.1=0
     trisb.2=0
     rem —-Accende 3 porte——-
     portb.0=1
     portb.1=1
     portb.2=1
     rem —-Azzera conteggio Duty Cycle——
     p1=0
     p2=0
     p3=0
     while true
          rem ——Incrementa i Tick——-
          p1=p1+1
          p2=p2+1
          p3=p3+1
          rem —Controlla Duty ON———
          if p1>80 then portb.0=0 end if
          if p2>30 then portb.1=0 end if
          if p3>10 then portb.2=0 end if
          rem —-Reinizia Intero periodo 100%———-
          if p1>100 then portb.0=1 p1=0 end if
          if p2>100 then portb.1=1 p2=0 end if
          if p3>100 then portb.2=1 p3=0 end if
          delay_us(100)
     wend
end.
```

### Tre PWM indipendenti tra loro

La generazione di due o più segnali PWM indipendenti tra loro è più difficoltosa. La filosofia di funzionamento è diversa dalla precedente.
Spieghiamo adesso in dettaglio come avviene la generazione dei vari segnali PWM, in modo indipendente e, soprattutto, in un unico ciclo infinito software.
Esaminiamo il **listato 3**. All'inizio del programma, i tre Led sono accesi. L'intero periodo del segnale (100%) viene idealmente suddiviso in 100 parti. Per i tre segnali PWM si utilizzano tre contatori indipendenti (p1, p2, p3) che incrementano di una unità il proprio valore a ogni iterazione del ciclo.
Se tali parametri raggiungono rispettivamente i valori di 80, 30 e 10, significa che il periodo ON è terminato e il livello logico della porta deve essere commutato su falso, fino al completamento dell'intero periodo, durante il quale il livello logico della porta deve restare a falso.

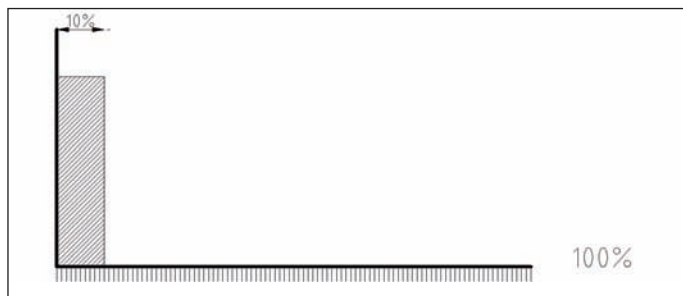



Figura 10: larghezza dei tre segnali PWM dell'80%, 30% e 10%

Infatti, non appena il contatore raggiunge il valore di 100, esso viene immediatamente azzerato, il LED si riaccende e il ciclo si ripete all'infinito.

Con questa tecnica si possono gestire agevolmente tanti segnali PWM software del tutto indipendenti tra di loro. Con i parametri adottati, un intero periodo ha una durata di 135 microsecondi, corrispondenti a una frequenza PWM di 7.407 Hz, valore molto elevato che evita l'osservazione del lampeggiamento dei LED. Misurando la tensione delle tre porte di uscita del PIC (RB0, RB1 e RB2) con un tester in continua, si possono rilevare le tensioni corrispondenti alle tre percentuali di duty cycle (80%, 30% e 10%) vale a dire 4 volt, 1,5 volt e 0,5 volt.
La misura deve essere sempre eseguita, ricordiamolo, senza il carico applicato. Ribadiamo anche che le porte non generano delle tensioni fisse, ma solo un treno di impulsi che il carico o il tester "vedono" come tensioni continue. Ovviamente l'utilizzo di un oscilloscopio dimostrerebbe proprio il contrario.

## CONCLUSIONI

Dopo avere affrontato le tecniche relative all'uso del segnale PWM di tipo software generato dai microcontrollori PIC vi diamo appuntamento alla prossima puntata, con la seconda parte dell'articolo, per trattare la generazione PWM via hardware. ❑

**CODICE MIP 2830023**

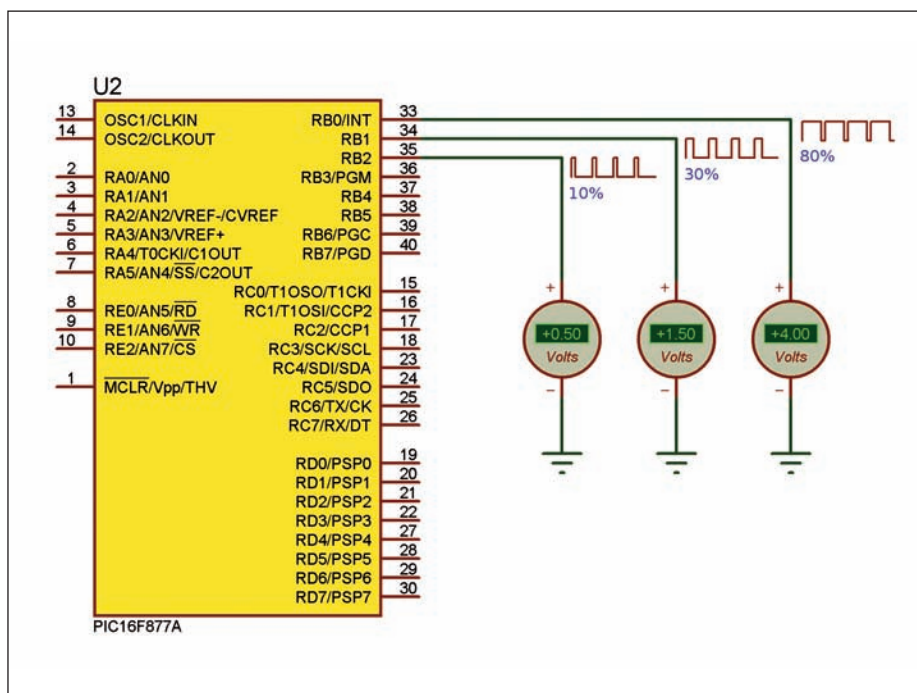



Figura 11: le tensioni di 4 volt, 1,5 volt e 0,5 volt generate dal PWM.

di GASPARE SANTAERA

# *Vasca per* PCB HOME MADE

**Realizziamo una vasca per l'incisione di PCB pratica ed economica, in grado di mantenere in temperatura il cloruro ferrico e, al tempo stesso, capace di tenere il fluido in movimento durante le fasi del processo.**

Chiunque abbia mai sviluppato una PCB, sa cosa significa cimentarsi con il cloruro ferrico e i rischi per un corretto sviluppo che l'utilizzo di quest'ultimo comporta. Un'errata o prolungata esposizione potrebbe rovinare o non ripulire a fondo la scheda.
D'altra parte chiunque si sia mai cimentato a utilizzare il cloruro ferrico sa che all'aumentare dell'energia cinetica del fluido aumenta l'efficacia di quest'ultimo riducendo quindi i tempi di incisione. Per queste ragioni molti consigliano di riscaldare il cloruro prima dell'utilizzo. Al tempo stesso molti utilizzano vasche basculanti, così da mantenere il fluido in movimento durante il processo. Da qui nasce l'idea di questo progetto, un vasca per l'incisione di PCB capace di mantenere il cloruro ferrico a una determinata temperatura e al tempo stesso capace di muovere il fluido durante il processo.
Per la costruzione ho usato per lo più materiali di riciclo. Il cloruro, come visibile in **figura 1**, si trova in una vecchia pirofila da forno, in vetro, capace di resistere al contatto anche a temperature superiori ai 100 °C. Questa è appoggiata a una piastra recuperata da un vecchio ferro da stiro, che si occupa di riscaldare la pirofila e di conseguenza il liquido al suo interno.

*Figura 1: vista d'insieme della vasca.*

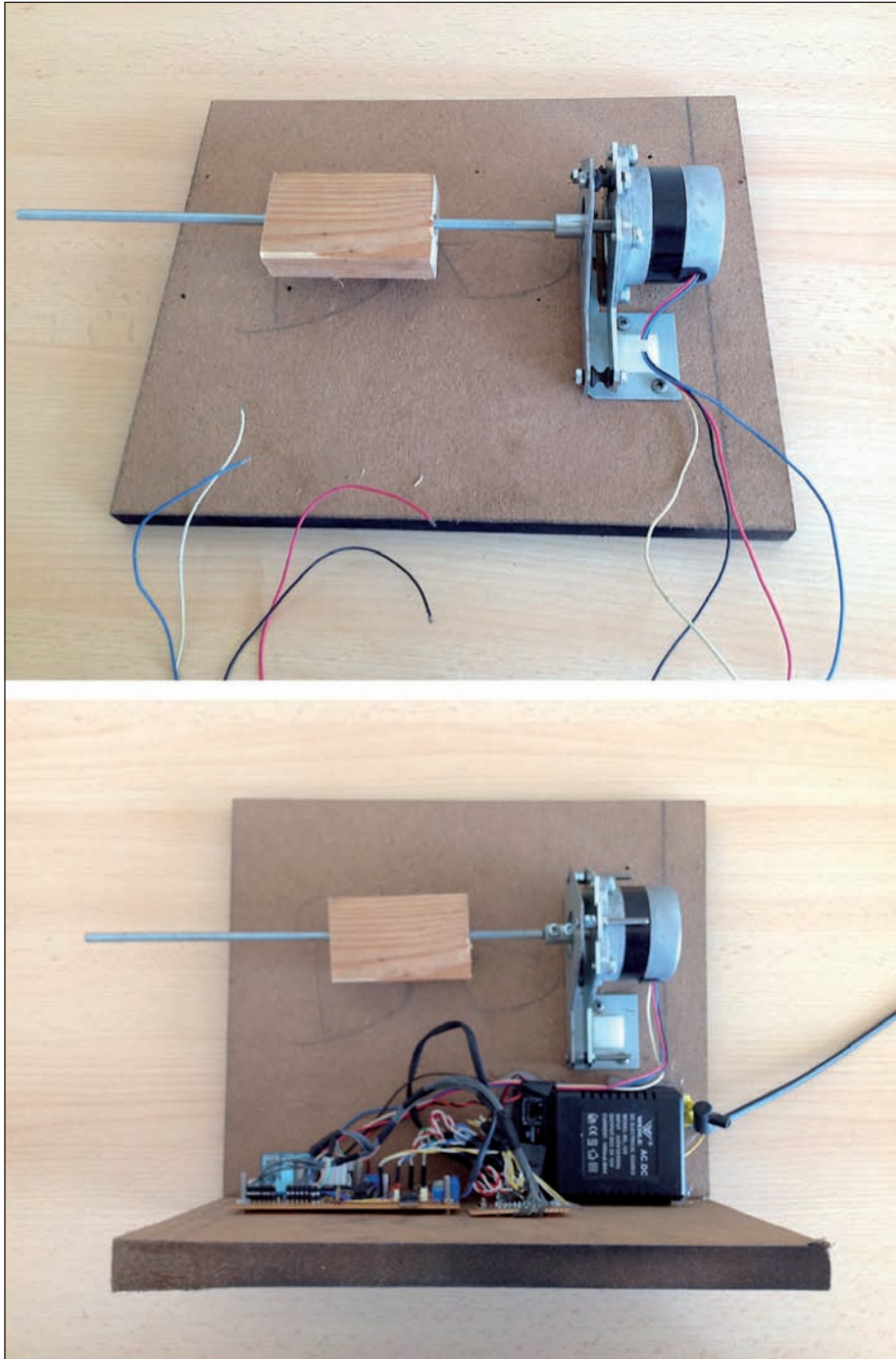



*Figura 2: fissaggio del motore sulla base e montaggio della scheda di controllo, del display e dell'alimentatore.*

Infine tale piastra si trova su un piano mobile, connesso a un altro piano contenente un distanziometro. Dentro il distanziometro ruota una barra filettata connessa rigidamente a un motore elettrico. Quando il motore ruota, ruota anche la barra. Di conseguenza il distanziometro trasla e con questo tutto il piano su cui si trova la piastra e quindi la pirofila. La gestione della temperatura del fluido e il controllo del motore elettrico è affidata un microcontrollore di tipo PIC16F876.

## COSTRUZIONE

Per la costruzione ho utilizzato prevalentemente pannelli di MDF, facili da model-

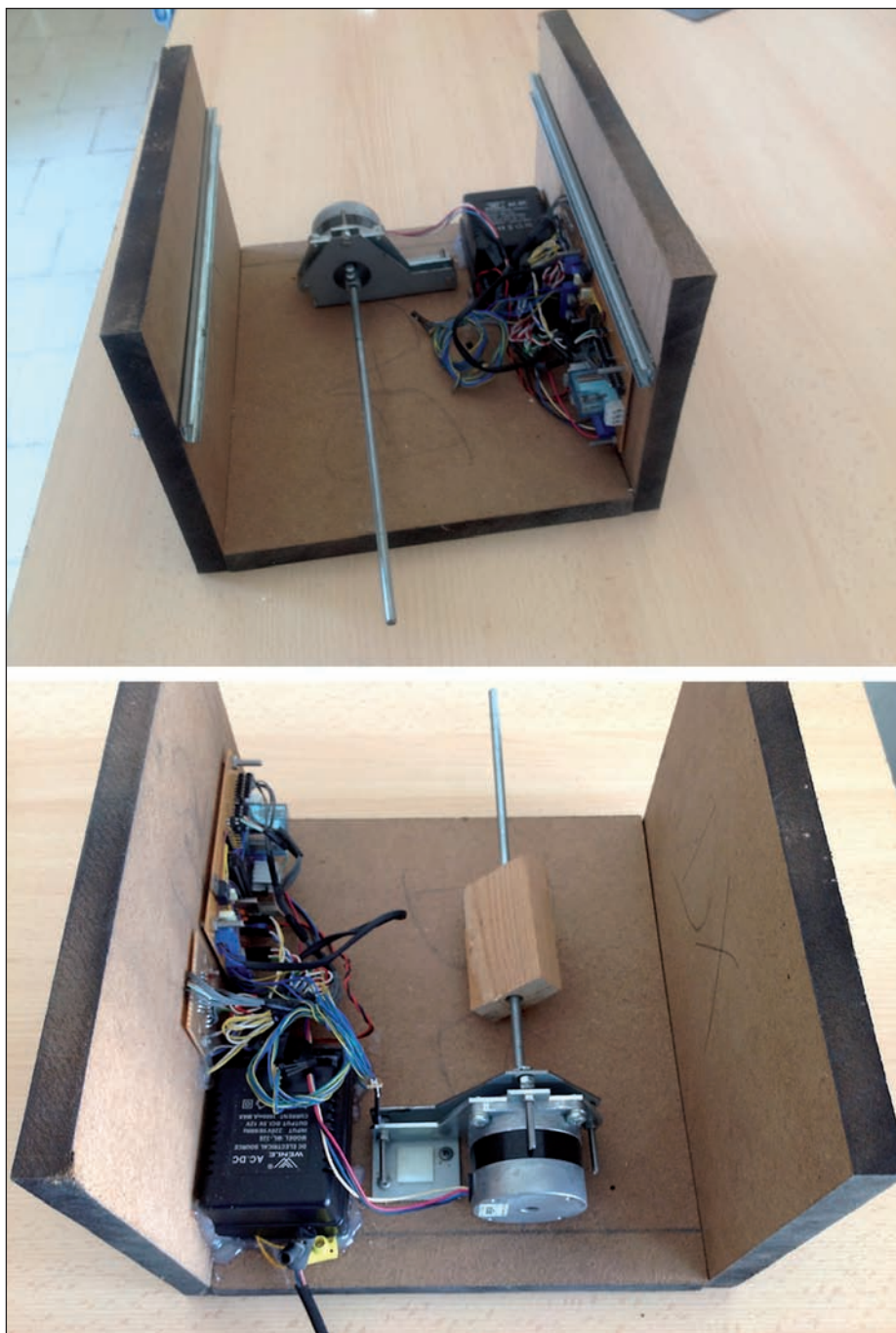



*Figura 3: montaggio della seconda faccia e montaggio delle guide per i carrelli.*

lare e al tempo stesso facili da lavorare. Per le dimensioni mi sono basato su quelle della pirofila, ovvero partendo dalla pirofila gli ho costruito la scatola attorno. Di seguito riporto le figure che descrivono passo-passo la costruzione.
Nei primi due passi **(figura 2)** si monta il motore, con relativa barra filettata, e l'asse che poi verrà collegato al piano della piastra, alla base. Successivamente si monta il lato frontale con l'elettronica, i display e i pulsanti. Per quanto riguarda l'alimentazione della scheda di controllo, ho preferito utilizzare un alimentatore universale da 1 A, in modo da evitarmene la costruzione la quale certamente mi avrebbe tolto tempo e spazio sulla base della vasca.
Nei passi 3 e 4 **(figura 3)** viene montato l'altro lato della vasca e le guide dove poi scorrerà il piano della piastra.
A questo punto si inserisce il piano che porta la piastra, fissandolo al piano dentro cui scorre la barra filettata e si collegano i sensori di temperatura, montati sulla piastra, alla scheda di controllo **(figura 4)**.
Infine si monta prima il lato posteriore con il relativo pulsante di accensione e il cavo di alimentazione e poi il lato anteriore con il relativo foro passante per la barra filettata **(figura 5)**.
Al link http://www.youtube.com/watch?v=EECwg9WC1TA&feature=plcp si trova un modellino 3D della costruzione.

## ELETTRONICA

Il cuore delle vasca è costituito da un microcontrollore, un PIC 16F876. Questo controlla un motore passo-passo tramite un driver, una coppia di sensori di temperatura, una piastra termica, e due display a sette segmenti.
Per quanto riguarda il motore passo-passo, questo viene utilizzato tramite un driver di tipo L298, il microcontrollore ha 4 pin appartenenti alla porta A, connessi al driver. Facendo costantemente variare i livelli di tensione tra questi si riesce in modo piuttosto facile e intuitivo a far ruotare il motore. Il controllo del motore viene attivato/disattivato da un interruttore posto sul lato frontale della vasca.
Per quanto riguarda i sensori di temperatura, sono stati utilizzati due LM335. Essi vengono montati nella configurazione di calibrazione così che, variando il valore del trimmer connesso in parallelo al singolo sensore, si può fissare la tensione d'uscita del sensore quando questo si trova a una temperatura nota, per esempio a 0 °C. I due sensori sono collegati ai piedini 0 e 1 della porta A, ovvero a due dei quattro convertitori analogico-digitali presenti nel microcontrollore.
La temperatura desiderata viene fissata mediante due dei quattro pulsanti presenti sulla vasca, gli altri due servono ad abilitare/disabilitare il controllo stesso. Una volta fissata la temperatura desiderata il microcontrollore la confronta con quella letta dai sensori e, nel caso, abilita la piastra portando a 1 il pin RC4. Questo accende il BJT e di conseguenza fa scattare il relè connesso alla piastra.

Figura 4: montaggio del piano porta piastra e dettagli del fissaggio del sensore di temperatura.

Infine per il controllo dei display sono stati utilizzati due shift register 74HC595, uno per display. Questa soluzione circuitale permette di gestire contemporaneamente due display a 7 segmenti utilizzando solamente 6 pin.
Il registro ha un ingresso seriale a 8 bit, un'uscita seriale a 8 bit e un'uscita parallela sempre a 8 bit, quest'ultima viene connessa al singolo display.
Il piedino MR del registro è il negato del master reset e viene collegato a un livello logico alto; il piedino OE è il negato dell'output enable input e viene collegato a massa in modo appunto da attivare l'uscita del registro.
Infine vengono utilizzati altri due piedini: SHCP e STCP i quali, insieme al piedino DS, ovvero l'ingresso seriale, si occupano di memorizzare il singolo bit e successivamente eseguire lo shift sui valori già presenti nel registro. Il meccanismo è molto semplice, quando bisogna memorizzare un nuovo valore nel registro. Partendo sempre dal bit più significativo, prima si porta il piedino del latch, ovvero SHCP a livello logico basso, il piedino DS uguale al bit che si sta scrivendo, e il piedino del clock ovvero STCP a livello alto, e poi si esegue lo shift a sinistra portando il piedino del latch a livello logico alto. Questa operazione va ripetuta per ogni bit che si intende scrivere. Nel caso in cui si usino tutti gli 8 bit, ad esempio, essa va eseguita otto volte.

## FIRMWARE

Il firmware è stato realizzato in mikroC. Il codice è molto semplice, dopo una prima fase di inizializzazione si entra in un ciclo while infinito. In quest'ultimo il programma gestisce la pressione dei pulsanti per l'aumento-decremento della temperatura desiderata e dei pulsanti che abilitano-disabilitano il controllo di temperatura.
Per quanto riguarda il controllo di temperatura, quando è attivo, il microcontrollore legge le tensioni provenienti dai due sensori e fa la media dei due. Considerando lo spessore del vetro presente tra il liquido e la piastra la temperatura percepita dai sensori viene abbassata di 5 °C, al fine appunto di approssimare l'attuazione termica dovuta al vetro.
A questo punto il microcontrollore confronta la temperatura percepita con quella desiderata, la quale è stata precedentemente convertita in bit. Nel caso in cui la temperatura desiderata sia superiore di 3 °C rispetto a quella percepita, viene attivata la piastra per 5 secondi e subito dopo disattivata, sempre per 5 secondi. Questo lasso di tempo è necessario affinché il sensore, che ha una dinamica più lenta della piastra, percepisca la giusta temperatura.
Nel caso in cui invece la temperatura percepita sia di 1 °C superiore a quella desiderata, la piastra viene mantenuta spenta.
Per quanto riguarda la conversione delle temperatura desiderata in bit, il sensore di temperatura è stato calibrato in modo da restituire 2,5 V quando si trova a 0 gradi e aumenta la sua tensione d'uscita

Figura 5: montaggio della parte posteriore e montaggio della parte anteriore con fissaggio della barra filettata.

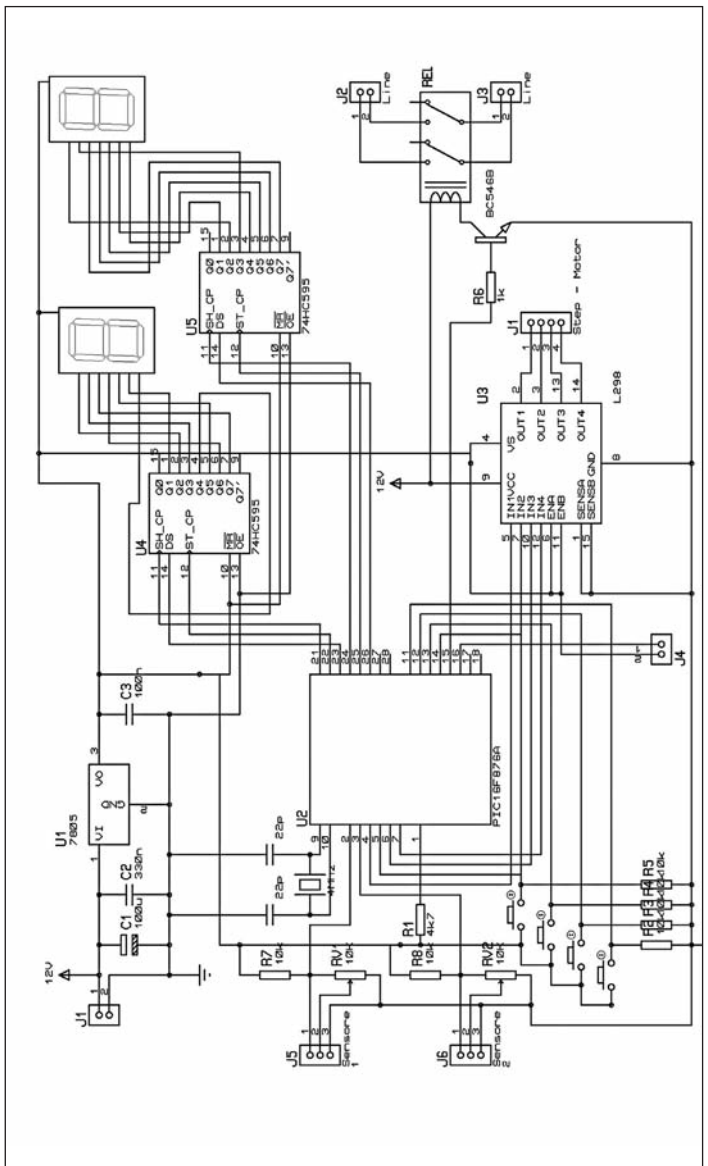



Figura 6: schema circuitale.

di 10 mV per ogni grado in più percepito. Avendo il convertitore ADC del micro una risoluzione di circa 5 mV, la variazione di 1 °C equivale circa a 2 bit, quindi la temperatura desiderata in bit sarà la somma di 512, ovvero 2,5 V in bit, più il valore di temperatura desiderato moltiplicato per 2.

Sempre nel ciclo while viene gestito il controllo dei due display a 7 segmenti che mostrano la temperatura desiderata. Ciclicamente viene chiamata un'apposita funzione (Display), la quale si occupa di gestire i pin connessi ai due registri utilizzati, uno per ogni display.

La funzione (Display) prende in ingresso la temperatura desiderata e ne calcola le decine e le unità, ognuna delle quali occuperà poi un apposito display.

Successivamente essa passa alla scrittura del valore delle unità nel primo registro, quello collegato appunto al display delle unità. Il registro utilizzato è un registro a scorrimento, quindi bisogna scrivere gli otto bit che può contenere partendo dal bit più significativo e procedendo di volta in volta con uno shift.

Il controllo del motore passo-passo avviene all'interno della funzione di interrupt (Interrupt) e, come già detto, ogni 2 secondi circa, essa implementa un contatore.

Successivamente la funzione controlla il valore numerico presente nel contatore: se trova un numero dispari chiama la funzione MotoreFW, che fa compiere al motore quattro giri in senso orario; se invece il contatore contiene un numero pari chiama la funzione MotoreRW, che viceversa fa compiere al motore quattro passi in senso antiorario.

In questo modo, quando il controllo del motore è attivo il programma ruota ogni 2 secondi la barra filettata in un verso e nell'altro. Ciò permette quindi di muovere il piano dove si trova la piastra, facendo muovere il cloruro presente all'interno in avanti e indietro senza mai sbattere sulle pareti interne della struttura. ❑

**CODICE MIP 2830371**

di ROBERTO D'AMICO (IW0GTF)

Tutorial PIC18 *(parte sesta)*

# COMUNICARE con ALTRI sistemi

**Quando un sistema raggiunge un certo grado di complessità, si avverte la necessità di comunicare con altri sistemi per scambiare informazioni.**

Se avete seguito le prime lezioni, pubblicate sui precedenti numeri di Fare Elettronica, ora avete le nozioni necessarie per interfacciare al PIC molti tipi di sensori e attuatori. Utilizzando questo tipo di periferiche si ha spesso la necessità di poter visualizzare le informazioni elaborate dal PIC in una forma chiara e leggibile per l'uomo. Ad esempio utilizzando il microcontrollore per gestire un sensore di temperatura è anche utile leggere il valore misurato. Per fare questo, uno dei possibili modi è trasmettere questa informazione verso un altro dispositivo, come potrebbe essere ad esempio un computer, tramite un'interfaccia seriale che utilizza lo standard RS-232.

## LA SERIALE

Questo tipo di comunicazione è un po' datato – infatti oggi la maggior parte dei computer portatili non possiede più una periferica di questo tipo – ma resta comunque uno standard molto usato, basti pensare che esistono integrati specializzati nella conversione da RS-232 a USB.
La comunicazione seriale tra dispositivi ha il grande vantaggio di utilizzare un numero ridotto di linee di I/O, sono sufficienti un canale di trasmissione e uno di ricezione nella modalità di funzionamento più semplice. I segnali che viaggiano attraverso il cavo di trasmissione hanno valori di tensione che variano da –15 volt a +15 volt, per indicare rispettivamente un bit a uno e un bit a zero. Di conseguenza si rende necessario l'utilizzo di uno specifico integrato per l'adattamento dei livelli tra la linea di trasmissione e il PIC, come ad esempio il MAX232. Quest'adattamento è necessario quando si deve collegare un circuito con un PC o un altro dispositivo che prevede la comunicazione con lo standard RS-232, ma può capitare che l'utilizzo di un integrato adattatore di livelli non sia necessario, ad esempio collegando due PIC tra di loro a breve distanza, oppure quando si utilizza qualsiasi dispositivo che prevede un funzionamento con livelli logici standard TTL (0-5 volt).

## IL CIRCUITO

Lo schema elettrico riprende, al solito, quello delle precedenti lezioni, al quale aggiungiamo solamente il circuito integrato MAX232 per adattare i livelli di trasmissione. I due segnali RX e TX del PIC, presenti rispettivamente sui pin 18 e 17, sono collegati ai pin 12 e 11 del MAX232 il quale, come componenti esterni, richiede solamente quattro condensatori elettrolitici da 1 microF.
Il circuito è poi collegato al connettore X1 a 9 poli, connettore standard per la comunicazione seriale su PC, che servirà per portare il segnale di TX del PIC verso l'ingresso RX del PC e viceversa.

## IL CAVO DI COLLEGAMENTO

Va posta particolare attenzione sul tipo di cavo utilizzato per il collegamento tra PIC e PC, in quanto esistono cavi che invertono internamente i due pin RX e TX. Se pensate di utilizzare questo tipo di cavi va cambiato il circuito di **figura 1** invertendo i due segnali di TX e RX tra il MAX232 e il connettore a 9 poli. Nella **figura 2** sono riportati i due schemi di collegamento tra il circuito e il PC. Nella parte alta è riportato il collegamento utilizzato per lo schema di **figura 1**, in cui il collegamento al PC è effettuato con un

*Figura 1: lo schema elettrico del circuito.*

cavo senza incrocio dei due segnali (come si può vedere il segnale trasmesso dal PIC arriva all'ingresso di ricezione del PC e viceversa). Nella parte bassa della figura è invece riportato l'esempio di collegamento con un cavo Null Modem, ovvero con i due segnali incrociati. In questo caso il circuito va modificato introducendo un ulteriore incrocio dei segnali tra l'adattatore dei segnali e il connettore. Questa operazione non fa altro che annullare l'incrocio del cavo.

Come accennato in precedenza, molti PC più recenti, in particolare portatili, non hanno porta seriale RS-232. Per ovviare a questo problema, esistono adattatori USB che possono essere utilizzati per collegare il circuito presentato in questa lezione al PC.

Per visualizzare sul computer i dati trasmessi dal PIC è necessario utilizzare un programma terminale, come ad esempio Tera Term per Windows. I parametri di trasmissione che utilizzeremo prevedono un baud rate di 9.600, dati da 8 bit, nessuna parità e un bit di stop.

## CONFIGURAZIONE MODULO EUSART

Il modulo utilizzato per la comunicazione seriale del PIC si basa su 5 registri di configurazione: TXSTA, RCSTA, BAUDCON, SPBRG e SPBRGH. I primi tre servono per la configurazione del modulo, mentre gli altri due rappresentano il registro a 16 bit utilizzato per la generazione del baud rate di trasmissione.

Per prima cosa si deve stabilire la velocità di trasmissione che si vuole utilizzare. Ipotizzando di utilizzare il classico valore 9.600 baud, sul data sheet troviamo la formula per calcolare il valore da usare per i registri SPBRG e SPBRGH:

```
((Fosc/BaudRateDesiderato)/4)-1
```

Considerando quindi un baud rate di 9.600 e la frequenza di funzionamento del PIC di 40 MHz abbiamo:

```
X=((40000000/9600)/4)-1=1040
```

Partendo dal valore calcolato, si può usare la formula inversa per stabilire il reale baud rate con cui avverrà la comunicazione:

```
Baud rate = Fosc/(4*(x+1) = 40000000/(4*(1040+1)) = 9606
```

Il valore ottenuto si avvicina moltissimo ai 9.600 baud richiesti, questo garantisce una frequenza molto precisa riducendo notevolmente anche la possibilità di errori durante la trasmissione.

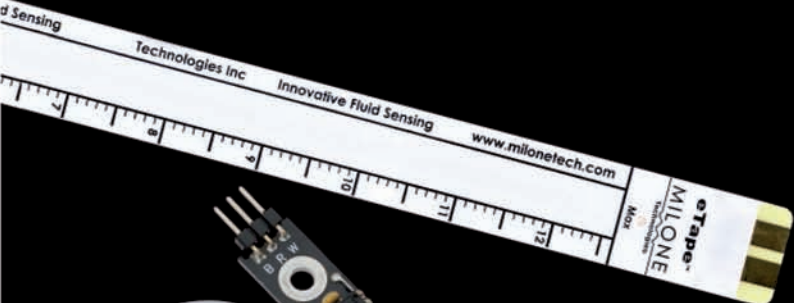
LISTATO 1

```
#include <p18f2620.h>
#include <delays.h>
/**********
 * DEFINE *
 **********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
 * MAIN *
 ********/
void main(void)
{
    // Impostazione dei registri di configu-
razione del PIC
    ADCON1=0b00001111;
    CMCON=0b00000111;
    // Configurazione Modulo UART
    SPBRG=0x10;
    SPBRGH=0x04;
    TXSTA=0b00100100;
    RCSTA=0b10010000;
    BAUDCON=0b00001000;
    // Reset iniziale delle porte
    PORTA=0;
    PORTB=0;
    PORTC=0;
    // Definizione della funzione delle sin-
gole porte
    TRISA=0x03;
    TRISB=0b00000011;
    // Configura RC7 come ingresso
    TRISC=0b10000000;
     // Accende il LED4 per indicare che il
programma è avviato
    LED1=1;
    LED2=1;
    LED3=1;
    LED4=1;
    Delay10KTCYx(250);
    LED1=0;
    LED2=0;
    LED3=0;
    LED4=0;
    // Trasmette il messaggio iniziale
    TXREG ='R';
    while(!TXSTAbits.TRMT);
    TXREG ='E';
    while(!TXSTAbits.TRMT);
    TXREG ='A';
    while(!TXSTAbits.TRMT);
    TXREG ='D';
    while(!TXSTAbits.TRMT);
    TXREG ='Y';
    while(!TXSTAbits.TRMT);
    // Loop infinito
    for(;;)
    {
        // Attende la ricezione di un Byte
        while(!PIR1bits.RCIF);
        // Accende il led in funzione del byte
ricevuto
        switch(RCREG)
        {
case '1': LED1=1;LED2=0;LED3=0;LED4=0;break;
case '2': LED1=0;LED2=1;LED3=0;LED4=0;break;
case '3': LED1=0;LED2=0;LED3=1;LED4=0;break;
case '4': LED1=0;LED2=0;LED3=0;LED4=1;break;
default : LED1=0;LED2=0;LED3=0;LED4=0;break;
        }
    }
}
```

Per popolare i due registri con il valore in precedenza calcolato, si deve convertire il valore in esadecimale, quindi 1040 corrisponde al valore 0x0410. Va quindi caricato come 0x10 in SPBRG e 0x04 in SPBRGH, rispettivamente la parte bassa e alta del registro a 16 bit. L'utilizzo dei due registri non è obbligatorio, ma sono utilizzati in questa lezione per avere una maggiore precisione della frequenza di trasmissione. Per abilitare l'utilizzo dei due registri è necessario settare il bit BRG16 del registro BAUDCON **(tabella 1)**.
Il registro TXSTA serve invece per configurare la parte di trasmissione del modulo. In particolare il bit TX9 a livello logico zero indica che la trasmissione sarà a 8 bit. Il bit TXEN abilita la trasmissione, SYNC serve invece per indicare la modalità di funzionamento Asincrona del modulo e infine BRGH, impostato a uno, indica la configurazione del modulo per alte velocità. Il bit TRMT è un bit di stato che può essere solo letto. Come avremo mo-

TABELLA 1

| BAUDCON | | | | | | | |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | - | WUE | ABDEN |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

*Tabella 1: registro BAUDCON e valori impostati.*

LISTATO 2

```
#define USE_OR_MASKS
#include <p18f2620.h>
#include <delays.h>
#include “usart.h”
/**********
 * DEFINE *
 **********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
 * MAIN *
 ********/
void main(void)
{
    char Txdata[]=”READY”;
    char Rxdata;
    /* Impostazione dei registri di configu-
razione del PIC */
    ADCON1=0b00001111;
    CMCON=0b00000111;
    /* Reset iniziale delle porte */
    PORTA=0;
    PORTB=0;
    PORTC=0;
    /* Definizione della funzione delle sin-
gole porte */
    TRISA=0x03;
    TRISB=0b00000011;
    /* Configura RC7 come ingresso */
    TRISC=0b10000000;
    /* Configurazione Modulo UART */
        OpenUSART(USART_TX_INT_OFF | US-
ART_RX_INT_OFF | USART_ASYNCH_MODE | US-
ART_EIGHT_BIT |
              USART_CONT_RX | USART_BRGH_HIGH
, 1040);
      baudUSART(BAUD_16_BIT_RATE | BAUD_AU-
TO_OFF);
     /* Accende il LED4 per indicare che il
programma è avviato */
    LED1=1;
    LED2=1;
    LED3=1;
    LED4=1;
    Delay10KTCYx(250);
    LED1=0;
    LED2=0;
    LED3=0;
    LED4=0;
    /* Attende che il modulo Usart sia libero
*/
    while(BusyUSART());
    /* Trasmette la stringa */
    putsUSART((char *)Txdata);
    // Loop infinito
    for(;;)
    {
        /* Attende la ricezione di un Byte */
        while(!DataRdyUSART());
        /* Legge il byte ricevuto */
        Rxdata=ReadUSART();
        /* Accende il led in funzione del byte
ricevuto */
        switch(Rxdata)
        {
case ‘1’: LED1=1;LED2=0;LED3=0;LED4=0;break;
case ‘2’: LED1=0;LED2=1;LED3=0;LED4=0;break;
case ‘3’: LED1=0;LED2=0;LED3=1;LED4=0;break;
case ‘4’: LED1=0;LED2=0;LED3=0;LED4=1;break;
case ‘5’: LED1=0;LED2=0;LED3=0;LED4=0;break;
        }
    }
}
```

do di vedere nei prossimi paragrafi, esso indica quando è terminata la trasmissione del dato **(tabella 2)**.
Il registro RCSTA riguarda invece la parte di ricezione del modulo, dove SPEN serve ad abilitare il modulo seriale, RX9 impostato a zero per indicare che in ricezione il dato gestito è a 8 bit e CREN per abilitare la ricezione **(tabella 3)**.
Per determinare quando è stata completata la ricezione di un byte si può controllare il flag degli interrupt RCIF del registro PIR1. In questa lezione non saranno comunque utilizzati gli interrupt, che vedremo in seguito. Per concludere la configurazione del modulo si deve impostare il pin RC7 come ingresso, in modo da ricevere correttamente quanto sarà trasmesso da altri dispositivi.

## PRIMA COMUNICAZIONE CON IL PC

Il primo programma che analizziamo invia al PC un messaggio che riporta la scritta “READY”. In seguito la pressione dei tasti numerici 1, 2, 3 e 4 del PC, determinerà l’accensione dei corrispondenti LED nel circuito. Premendo qualsiasi altro tasto alfanumerico i LED saranno spenti.
Nella prima parte del **listato 1** troviamo la configurazione dei registri, di cui abbiamo parlato nel paragrafo precedente, e la configurazione del bit 7 della porta C come ingresso, tramite il registro TRISC.
Al termine della configurazione dei registri troviamo l’ormai classico lampeggio dei quattro LED per indicare che il programma è stato avviato. Subito dopo inizia la trasmissione del primo messaggio ver-

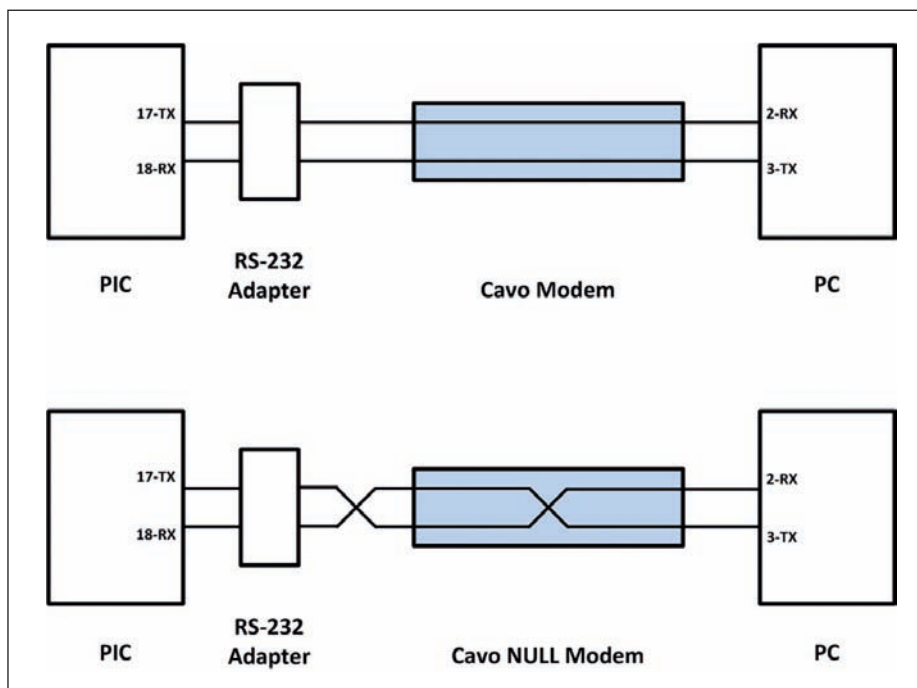



*Figura 2: lo schema di collegamento con i due tipi di cavo.*

TABELLA 2

| TXSTA | | | | | | | |
|---|---|---|---|---|---|---|---|
| **CSRC** | **TX9** | **TXEN** | **SYNC** | **SENDB** | **BRGH** | **TRMT** | **TX9D** |
| **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** |

*Tabella 2: registro TXSTA e valori impostati.*

so il PC. Per trasmettere un byte tramite il modulo EUSART si utilizza il registro TXREG il quale nel **listato 1** è valorizzato con i caratteri “R”, “E”, “A”, “D” e “Y”, che sarà il messaggio che vedremo sul programma terminale del PC. Dopo ogni caricamento del registro TXREG troviamo un’istruzione while che esegue il controllo sul bit TRMT del registro TXSTA. Com’è facile intuire questo controllo serve ad attendere che il byte sia stato trasmesso, perché questo bit (che può essere solo letto) passa a livello logico alto solo quando il registro TXREG è vuoto.
Trasmesso il primo messaggio al PC, il programma passa al loop infinito determinato dal ciclo FOR, nel quale un’altra istruzione while attende la ricezione di un byte. Questa volta il controllo avviene tramite il bit RCIF del registro PIR1 che, come avremo modo di vedere nelle prossime lezioni, è un registro per la gestione degli interrupt, anche se questo bit di controllo può essere utilizzato anche senza l’utilizzo degli interrupt.
A questo punto il byte ricevuto dal PC è memorizzato nel registro RCREG, il cui valore è gestito dall’istruzione switch. Questa nuova istruzione è utilizzata quando ci si trova nella condizione di dover gestire una serie di scelte multiple e il valore determinato dall’espressione definita all’interno delle parentesi che seguono lo switch servirà a stabilire quale delle istruzioni case sarà eseguita. Il case è il punto di controllo dell’espressione, se a questa è associato lo stesso valore determinato dall’espressione sarà eseguito il codice che segue fino al raggiungimento della prima istruzione break.
Nel codice del **listato 1** abbiamo che l’espressione è rappresentata dal registro RCREG, che contiene il dato ricevuto, mentre i primi 4 case indicano il codice da eseguire nel caso il PIC ricevi dal PC il carattere 1, 2, 3 o 4. L’ultima riga all’interno dello switch effettua un altro controllo default, questo vuol dire che per tutti gli altri valori dell’espressione che non sono gestiti dal case sarà eseguito il codice che segue il controllo default.
Come potete vedere per ogni case sono state scritte le istruzioni necessarie per l’accensione del solo LED corrispondente al tasto premuto, mentre nel default sono spenti tutti i LED. Altra nota è la presenza dell’istruzione break, a titolo informativo va detto che se non viene aggiunta questa istruzione al termine del codice del case, il programma eseguirà anche le istruzioni del case successivo e così via, fino al punto in cui incontrerà un break o il termine dello switch.

## LE FUNZIONI DEL C18 PER LA GESTIONE DEL MODULO EUSART

In questo paragrafo vediamo ora come utilizzare le funzioni del C18 per gestire il modulo EUSART **(listato 2)**. Come abbiamo visto per altre periferiche, per prima cosa è necessario includere nel programma il file che in cui sono definite, ovvero usart.h.
Per prima cosa vediamo la definizione di due variabili Txdata e Rxdata, la prima delle quali rappresenta un array. Un array è sostanzialmente un aggregato di variabili dello stesso tipo, cui si può accedere con un unico nome e un indice. Questo indice è indicato nelle parentesi quadre che seguono il nome della variabile. Nella definizione di Txdata si sta indicando, tramite le parentesi quadre aperte e chiuse, che la variabile rappresenterà un array di char, che è valorizzato con la stringa “READY”. Ciò vuol dire che la variabile avrà una dimensione di cinque byte e che in ognuno di essi sarà memorizzato un carattere della stringa. Avremmo quindi che accedendo alla posizione 2 dell’array, tramite l’istruzione Txdata[1] (il numero 1 all’interno delle parentesi quadre è corretto perché la numerazione parte da 0) questa conterrà il carattere ‘E’.
L’inizializzazione della periferica è affidata alle due funzioni OpenUSART e BaudUSART. Nella prima vanno indicati i parametri di configurazione del modulo e il valore da inserire nei registri SPBRG, mentre nella seconda è indicato l’utilizzo del regi-

TABELLA 3

| RCSTA | | | | | | | |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

*Tabella 3: registro RCSTA e valori impostati.*

stro a 16 bit per la temporizzazione dei dati da trasmettere e la disattivazione del sistema per la determinazione automatica del baud rate. Prima di eseguire la trasmissione dei dati verso il PC troviamo l'istruzione while che esegue il controllo tramite l'istruzione BusyUSART, per verificare che non ci sia una trasmissione in corso. L'istruzione che segue putsUSART prende come parametro l'array Txdata che contiene la stringa da inviare al PC. Come potete vedere è utilizzato il cast alla variabile con un asterisco il quale sta a indicare che la variabile è un array. Notiamo anche che la funzione inizia con la parola "puts" che è plurale, il che vuole appunto indicare che la funzione può inviare più byte e, per questo motivo, necessità di una struttura dati che può contenere più valori di quel tipo, per l'appunto un array. Grazie a questa istruzione non dovremo caricare ogni singolo byte nel registro TXREG come nell'esempio del **listato 1**, ma la funzione stessa trasmetterà tutti i byte contenuti nell'array, semplificando la scrittura del nostro programma. All'interno del loop troviamo, anche in questo caso, l'istruzione while la cui espressione è definita dalla funzione DataRdyUSART. Questa funzione esegue lo stesso controllo del **listato 1** sul bit RCIF per attendere che sia stato ricevuto un byte. L'istruzione successiva è la chiamata alla funzione ReadUSART, la quale restituisce il byte ricevuto che è memorizzato nella variabile Rxdata. Come per il paragrafo precedente l'istruzione switch è utilizzata per determinare quale LED deve essere acceso.

## CONCLUSIONI

Abbiamo visto in questa lezione un semplice approccio alla comunicazione seriale tra PIC e il PC. Ma questo non è l'unico modo per cui può essere utilizzata, esistono anche sensori che possono essere controllati in questo modo, oppure moduli che trasformano la trasmissione seriale in altri tipi di comunicazione, come la USB o comunicazioni wireless come xBee, Bluetooth e Wi-Fi. Negli esempi della lezione abbiamo messo in pratica il modo più semplice di utilizzo del modulo EUSART del PIC ma, come avrete visto, ci sono altri parametri che permettono configurazioni diverse, una fra tutte la possibilità di instaurare comunicazioni Master/Slave che, tramite una specifica gestione, permette di creare una piccola rete con più dispositivi. Ma qui si va oltre gli scopi di questo tutorial.

Per fare qualche esperimento con l'istruzione switch provate a togliere i break da tutti i case tranne che per default e per il case 4, cercando di capire che cosa sta succedendo. ❏

**CODICE MIP 2821445**

*Figura 3: il circuito montato sulla bread-board.*

Corso di Mikropascal
*(parte terza)*

# GESTIONE degli I/O

**In questa terza puntata del corso approfondiremo l'I/O di stati logici. In particolare vedremo come acquisire stati logici da cui far dipendere il flusso di programma. Tratteremo quindi, con alcuni esempi applicativi, le istruzioni condizionate IF-THEN-ELSE, l'istruzione di ciclo FOR e alcune funzioni predefinite di delay che spesso risultano utili nelle applicazioni.**

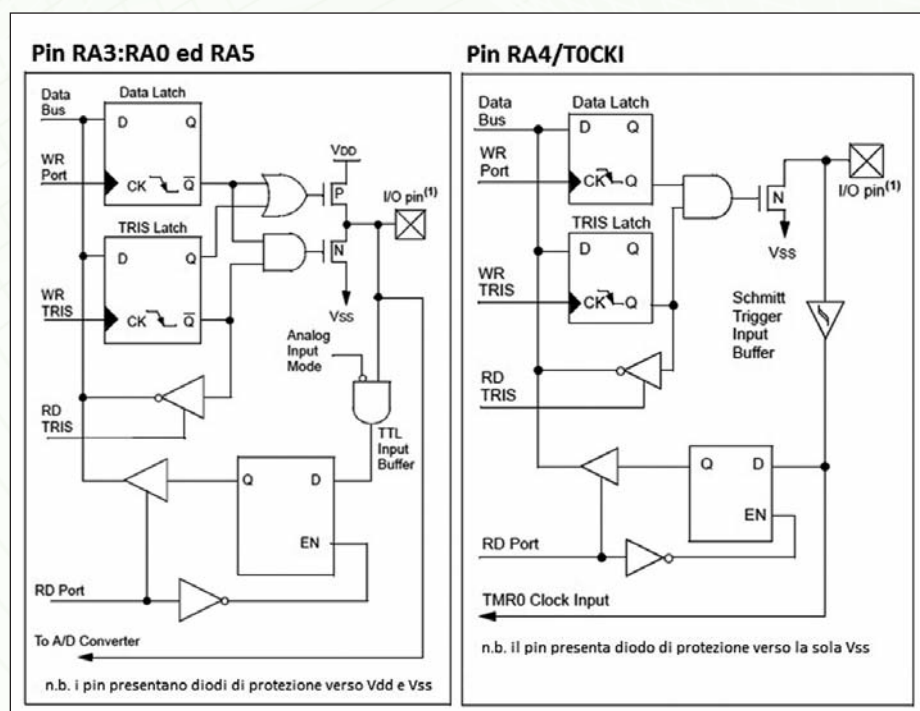
Nella seconda puntata del corso MikroPascal per PIC abbiamo approfondito la conoscenza pratica dell'IDE MikroPascal utilizzandolo per la creazione dei nostri primi progetti. Abbiamo in questo modo imparato come creare un nuovo progetto, come utilizzare alcune funzioni dell'editor che facilitano la stesura del codice, abbiamo preso confidenza con la struttura generale di un codice Pascal e soprattutto abbiamo imparato a configurare e utilizzare le porte del PIC come uscite impiegando istruzioni MikroPascal.
Abbiamo anche imparato a utilizzare, con una certa disinvoltura, dati espressi in notazione decimale, esadecimale e binaria, a compilare il codice di programma e a riconoscere gli errori più banali che possono far fallire la compilazione e il linking del firmware.
In questa terza puntata approfondiremo l'I/O di stati logici. In particolare, vedremo come programmare il PIC in maniera da acquisire stati logici in ingresso. Per rendere la trattazione degli argomenti più esaustiva, analizzeremo un po' più da vicino la circuiteria interna delle porte di I/O in modo da non lasciare l'aspetto della loro programmazione firmware del



*Figura 1: diagramma a blocchi della circuiteria interna della porta A di I/O del PIC16F877. Si noti la necessità di pull-up per il pin RA4 (a causa dell'open-drain del mosfet), nel caso in cui questo venga impiegato come uscita.*

di ANTONIO GIANNICO

tutto slegata dagli aspetti hardware e rendere il lettore e lo sperimentatore più consapevoli del nesso che esiste tra le due cose.
Vedremo come sia possibile implementare una macchina combinatoria o sequenziale in firmware il cui comportamento sia dipendente dal valore di stati logici acquisiti in ingresso al controllore. Nel trattare l'argomento avremo modo di sottoporre all'attenzione del lettore, con alcuni esempi applicativi, le istruzioni condizionate IF-THEN-ELSE, l'istruzione di ciclo FOR e alcune funzioni predefinite di delay che risultano spesso utili nelle applicazioni pratiche dei controllori.

## CONFIGURAZIONE E IMPIEGO DELLE PORTE DI I/O COME INGRESSI DI STATI LOGICI

Negli esempi proposti nella scorsa lezione non abbiamo mai utilizzato le porte del controllore come ingressi ma piuttosto come uscite. Un sistema che esegue delle operazioni senza percorrere il flusso di processo in base alla "percezione" e quindi all'analisi di segnali provenienti dall'esterno non è, in effetti, un dispositivo intelligente di I/O poiché non può controllare realmente ed efficacemente un processo secondo una logica di feedback. Un controllore è invece per sua natura qualcosa che è potenzialmente in grado di interagire con il mondo esterno secondo uno specifico algoritmo firmware che implementa una macchina combinatoria o sequenziale. Nel primo caso le uscite in ciascun istante dipendono esclusivamente dai valori assunti dagli ingressi in quello stesso istante; nel secondo caso invece le uscite in un determinato istante dipendono non solo dal valore degli ingressi in quell'istante ma anche dallo stato pregresso della macchina e quindi dalla sua storia. La macchina viene cioè ad avere un comportamento con memoria. Questi aspetti appariranno evidenti a partire da questa lezione poiché da ora in poi avremo modo di programmare in MikroPascal le porte del PIC utilizzandole sia come ingressi che come uscite. Riusciremo, in questo modo, a utilizzare il controllore come strumento in grado di elaborare degli stati logici o dei dati acquisiti in ingresso al fine di pilotare altri circuiti in uscita o di fornire risultati che scaturiscono proprio dall'elaborazione degli ingressi.

## L'HARDWARE INTEGRATO DELLE PORTE DI I/O

Prima di trattare la configurazione firmware delle porte del PIC utilizzate come ingressi è opportuno, per avere una visione globale dell'argomento, analizzare brevemente alcuni aspetti della circuitistica delle porte di I/O. Riusciremo in questo modo a comprendere meglio perché e in che modo sia possibile, da un punto di vista hardware, utilizzare le porte del PIC come ingressi piuttosto che come uscite. A tale scopo analizziamo lo schema a blocchi circuitale delle porte A, B, C, D e E del PIC 16F877.

### Porta A

Tutti i pin RAx possono essere utilizzati, con le medesime considerazioni, come ingressi o uscite logiche 0-5 V a meno di RA4, per il motivo che apparirà più chiaro tra breve. Lo schema del controllo hardware integrato dei pin RA0...RA3 e RA5 **(figura 1)** può essere scomposto in un blocco di ingresso composto da un Buffer TTL che si pone verso l'esterno a impedenza elevata e un FF D, e un blocco di uscita composto da uno stadio di uscita CMOS (due mosfet accoppiati p ed n rispettivamente) a sua volta pilotato da una circuiteria a monte.
Ovviamente, se tale stadio è spento (mosfet spenti) il pin funziona come ingresso attraverso il buffer TTL o a trigger di Schmitt (pin RA4), mentre se è acceso funziona come pin di uscita con uscita logica 0 o 1 a seconda dei valori logici che la circuiteria a monte determina sui gate dei due mosfet. Il fatto che la circuiteria abiliti il pin a fungere da ingresso piuttosto che da uscita dipende dai bit contenuti in uno speciale registro di configurazione delle porte detto registro di direzione (direction register), un particolare SFR (Special Function Register) che nel caso della porta A prende il nome di TRISA.
La porta A può, più in generale, essere vista come una porta a sei bit ciascuno dei quali corrisponde a uno specifico pin di I/O. Il registro di direzione TRISA è un registro anch'esso ampio 6 bit. Ciascun bit corrisponde a un pin di I/O della porta. Se tale bit del registro TRISA è 1 il pin risulta configurato, da un punto di vista hard-

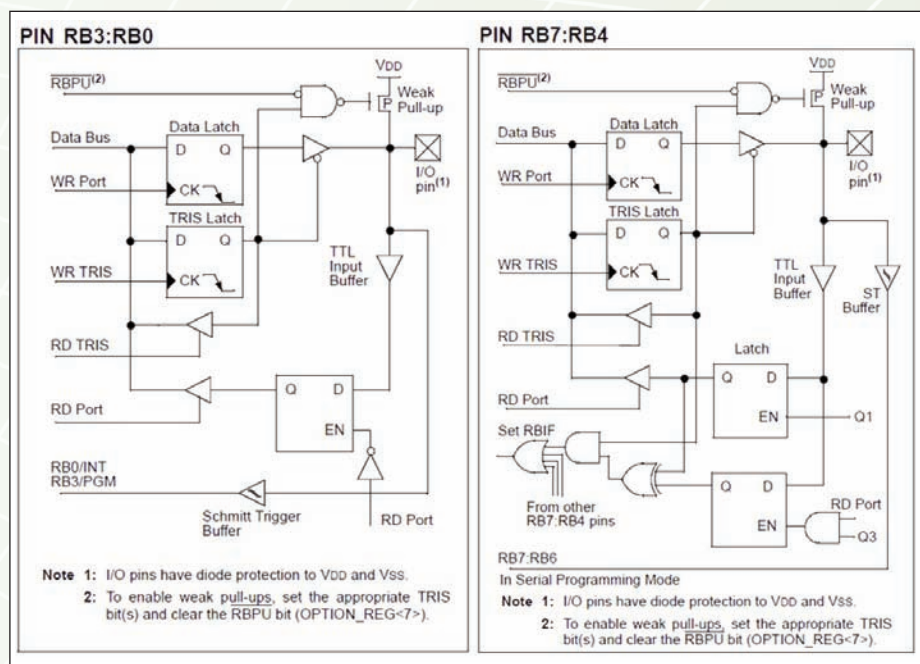



*Figura 2: diagramma a blocchi della circuiteria interna della porta B di I/O del PIC16F877.*

ware, come ingresso; se invece è 0, al contrario risulta configurato come uscita. Della porta A merita una considerazione a parte il pin di I/O RA4. Infatti, l'uscita presenta il solo Mosfet a canale N; si tratta in altre parole di uno stadio di uscita open-drain **(figura 1)**. Ciò comporta che se si vuole pilotare correttamente questo pin come uscita è necessario porre su di esso una resistenza di pull-up verso l'alimentazione VDD, facendo attenzione al valore della stessa. Infatti, un valore eccessivamente piccolo porterebbe il mosfet n a essere interessato da una corrente eccessiva in caso di uscita 0. Se si ipotizza di non superare 10 mA di corrente, è necessario utilizzare una resistenza di valore non inferiore a 5 V / 10 mA = 500 ohm. Nella pratica utilizzeremo una resistenza di alcuni chiloohm. Non ci soffermiamo per il momento sulle altre funzioni di questi pin che non siano di semplice I/O di stati logici. Ci limitiamo a segnalare che qualunque altra funzione multiplexata viene imposta programmando in firmware specifici bit di specifici registri del controllore. Per maggiori dettagli su quanto esposto si rimanda ai paragrafi:

- 3.0 - I/O PORTS;
- 3.1 - PORTA and the TRISA Register;

del documento **[1]** liberamente scaricabile dalla Rete e in particolare dal sito di Microchip **[2]**.
Per il momento ci limiteremo quindi a vedere la porta A come una porta a 6 bit che consente di pilotare, indipendentemente, altrettanti pin di uscita o di acquisire, indipendentemente, altrettanti livelli logici in ingresso.

## Porta B

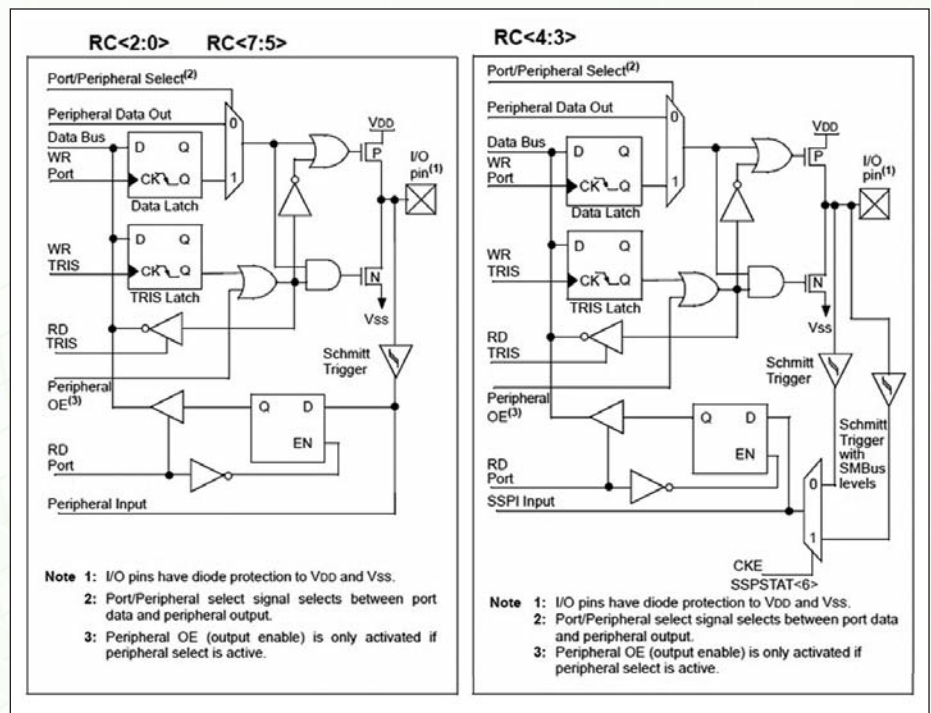
La porta B è anch'essa una porta bidirezionale, ma a 8 bit; consente per questo di pilotare indipendentemente altrettanti pin di uscita oppure di acquisire sugli stessi pin indipendentemente altrettanti livelli logici in ingresso. La configurazione di ciascun pin come ingresso o come uscita è determinata dai valori assegnati ai corrispondenti bit del registro TRISB. Se a uno specifico bit del registro TRISB viene assegnato valore 1 logico, il relativo pin della porta B funziona da ingresso; se invece allo stesso bit viene assegnato valore 0 logico, il relativo pin della porta funziona come uscita.



*Figura 3: diagramma a blocchi della circuiteria interna della porta C di I/O del PIC 16F877.*

La diversa circuiteria dei pin della porta B **(figura 2)** è legata alle specifiche funzioni multiplexate su di essi. Per esempio, i pin RB3/PGM, RB6/PGC e RB7/PGD presentano doppio acronimo in quanto utilizzabili sia come I/O in fase di running sia come pin di programmazione durante la fase di scrittura della memoria flash. RB3, in particolare, consente la funzione LVP (Low Voltage Programming), funzione che deve essere disabilitata in firmware se si vuole che dopo la programmazione il pin possa essere utilizzato come normale pin di I/O. La struttura circuitale interna è leggermente differente da quella della porta A, ma in ogni caso, se il pin è configurato come ingresso, il mosfet a canale p indicato come weak pull-up viene messo in stato di off per cui la sezione di uscita è come se non ci fosse; in queste condizioni entra in gioco l'input buffer TTL.
Una nota a parte meritano i pin RB4…RB7 che integrano anche funzioni di interrupt on change cioè la caratteristica di poter generare un interrupt in corrispondenza di una variazione dello stato logico del segnale che insiste come ingresso su di essi. Ciò ne spiega la differente circuiteria. Come RB4…RB7, anche RB0 è in grado di intercettare un evento esterno generando un interrupt. Avremo modo di approfondire questo argomento durante il corso. Per ulteriori approfondimenti è, per il momento, possibile consultare il paragrafo 3.2 (PORTB and the TRISB Register) di **[1]**.

## Porta C

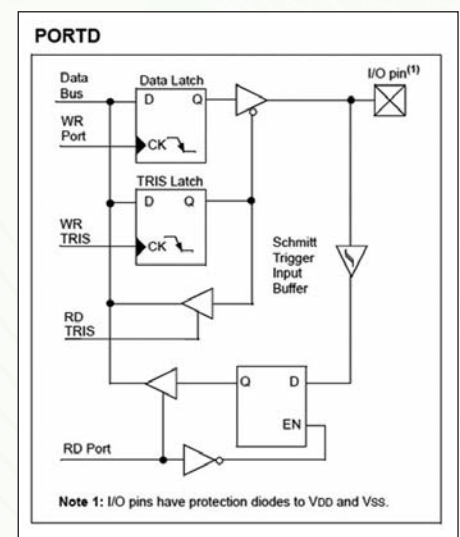
La porta C è anch'essa una porta bidire-



*Figura 4: diagramma a blocchi della circuiteria interna della porta D di I/O del PIC16F877.*

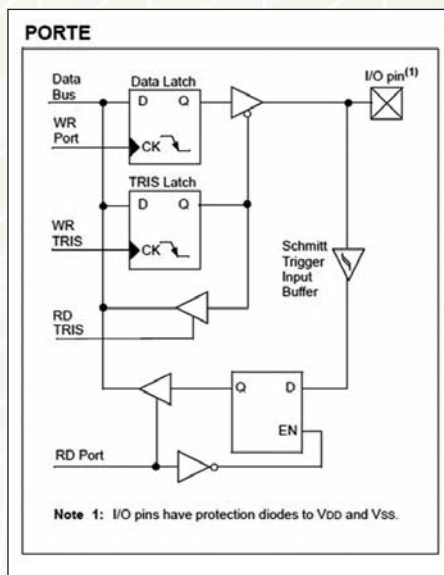



*Figura 5: diagramma a blocchi della circuiteria interna della porta E di I/O del PIC 16F877.*

zionale a 8 bit che consente di pilotare, indipendentemente, altrettanti pin di uscita o di acquisire in ingresso sugli stessi pin, indipendentemente, altrettanti livelli logici. La configurazione come ingresso o uscita di ciascuno di questi pin è determinata dai valori assegnati ai corrispondenti bit del registro di direzione TRISC. Ancora una volta, i particolari della circuiteria **(figura 3)** dipendono dal fatto che sugli stessi pin sono multiplexate differenti funzioni di I/O (tra cui funzioni PWM, I$^2$C e USART, di cui avremo modo di parlare in puntate successive del corso). Per maggiori dettagli è possibile consultare il paragrafo 3.3 (PORTC and the TRISC Register) di **[1]**.

*Figura 6: listato del programma di Esempio n. 1 (IO-ver.a).*

## Porta D

Anche la porta D **(figura 4)** è bidirezionale e di ampiezza 8 bit. La sua particolarità, rispetto alle altre, sta nel fatto di poter essere configurata come parallel slave port. La direzione è controllata mediante il solito registro di direzione che in questo caso prende il nome di TRISD. Per maggiori dettagli si rimanda alla lettura del paragrafo 3.4 (PORTD and TRISD Registers) di **[1]**.

## Porta E

La porta E **(figura 5)** è una porta di soli tre bit i cui pin possono essere utilizzati come pin di I/O di livelli digitali oppure come uscite analogiche. Valgono le solite considerazioni sul registro di direzione (che in questo caso prende il nome di TRISE) per quanto riguarda la configurazione dei pin come ingressi piuttosto che come uscita. Per maggiori dettagli è possibile consultare il paragrafo 3.5 (PORTE and the TRISE Register) di **[1]**.

Quanto esposto sulle porte di I/O è complesso nei dettagli circuitali che qui abbiamo volutamente descritto in maniera semplificata; in realtà le porte di I/O vengono utilizzate con estrema semplicità grazie al fatto di poter essere configurate e programmate attraverso un linguaggio di programmazione ad alto livello, cosa che consente di tenere nascosta al programmatore una buona parte dei dettagli circuitali e architetturali del controllore. Questa rapida carrellata sulla circuiteria delle porte di I/O è in ogni caso utile in quanto consente di avere un'idea concreta di come esse siano implementate da un punto di vista circuitale e perché.

Anche se non indicati esplicitamente negli schemi mostrati finora, tutti i pin di I/O sono protetti da diodi sia verso Vss sia verso Vdd, a eccezione di RA4 che risulta protetto unicamente verso Vss per ovvi motivi (mosfet in open drain). Chiaramente, questo non significa che sia possibile porre su questi pin qualunque segnale; se per esempio ponessimo su un pin una tensione di 5 V quando lo stesso sta funzionando da uscita e presenta livello logico 0, con tutta probabilità danneggeremmo il mosfet n dello stadio di uscita; se invece ponessimo su di esso 20 V quando lo stesso sta funzionando da ingresso allora l'ingresso potrebbe non riportare conseguenze se tale tensione è applicata attraverso una resistenza, mentre invece potrebbe portare al corto del diodo di protezione nel caso in cui sia applicata direttamente.

Facciamo in ultimo osservare come il multiplexing di più funzioni sugli stessi pin di un chip è una scelta progettuale piuttosto ricorrente nell'hardware dei dispositivi programmabili; basti pensare a quanti pin lo stesso controllore dovrebbe avere se non si usasse il multiplexing delle funzioni.

Ovviamente, tutto questo va a scapito

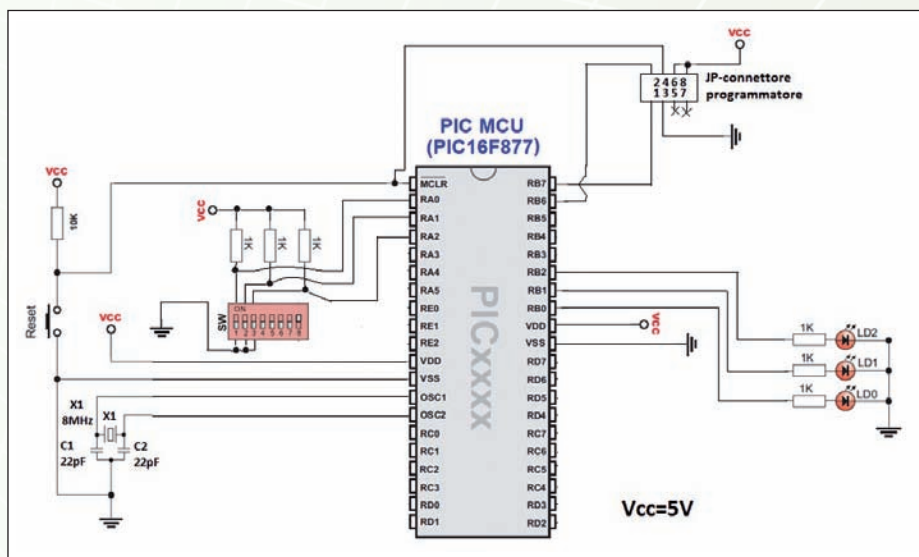
*Figura 7: schema circuitale del codice di esempio di figura 6 (esempio n. 1: IO-ver.a).*

```
if (condizione) then
                Begin
                  Istruzione_1;
                  Istruzione_2;
                  ....
                  Istruzione_n;
                end;
```

*Figura 8: sintassi generalizzata del costrutto IF-THEN senza ramo ELSE.*

```
if (condizione) then Istruzione_1
                  else
                     Istruzione_2;
```

*Figura 9: sintassi del costrutto IF-THEN-ELSE.*

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 3
 * Nome progetto: IO -ver. b - Revisione: 2012-10-06
 * Autore: A. Giannico
 * Descrizione:
     Programma è del tutto equivalente ad IO ver. a. Mostra come due istruzioni
     IF-THEN con condizioni mutuamente esclusive siano di fatto equivalenti ad
     una sola istruzione IF-THEN ELSE. A tale scopo si confronti il listato con
     quello di IO-ver. a
 *   MCU:PIC18F877; Scheda di sviluppo:Easypic3; Oscillatore: HS 8.0000 MHz Crystal
     SW: mikroPascal PRO for PIC  ver. 5.61
     http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
  }
program IO_b;
{ Dichiarazione variabili }
begin
  // configurazione della direzione dei singoli pin di I/O
  TRISA.0 := 1;  // A0, A1, A2 vengono configurati come ingressi
  TRISA.1 := 1;
  TRISA.2 := 1;
  TRISB.0 := 0;  // B0, B1 e B2 vengono configurati come uscite
  TRISB.1 := 0;
  TRISB.2 := 0;
  while TRUE do
    begin
     if (PORTA.0=1) then PORTB.0:=1  // equivalente a
                    else PORTB.0:=0; // if (PORTA.B0=1) then PORTB.B0:=1;
     if (PORTA.1=1) then PORTB.1:=1  //                   else PORTB.B0:=0;
                    else PORTB.1:=0;
     if (PORTA.2=1) then PORTB.2:=1
                    else PORTB.2:=0;
    end;       // Fine Main program
end.
```

*Figura 10: esempio n. 2 (IO-ver.b).*

```
if (condizione) then
                Begin
                  Istruzione_1;
                  Istruzione_2;
                  ...
                  Istruzione_n;
                end
                  else
                Begin
                  Istruzione_n+1;
                  Istruzione_n+2;
                  ...
                  Istruzione_n+m;
                end;
```

*Figura 11: costrutto IF-THE-ELSE applicato a blocchi di istruzioni mutuamente esclusivi.*

della complessità della progettazione dell'hardware del chip ma a enorme vantaggio della sua compattezza con conseguenti vantaggi in termini di costo e di compattezza di tutte le applicazioni nelle quali il componente verrà utilizzato.

## INTERFACCIAMENTO DI CONTATTI

Abbiamo già visto, nella precedente puntata, come creare un nuovo progetto per cui non ci dilunghiamo sui diversi step che diamo ormai per acquisiti e passiamo direttamente ad alcuni semplici codici di esempio che ci consentiranno di introdurre i primi costrutti del linguaggio MkroPascal. Il PIC che utilizziamo è ancora una volta il PIC16F877 con frequenza di clock fissata a 8MHz attraverso il solito circuito quarzato.
Per eseguire la compilazione utilizziamo il comando Build dell'omonimo menu a tendina oppure la combinazione di tasti Ctrl+F9 **(figura 11)**. La compilazione del codice MikroPascal produce come output il codice eseguibile .hex.

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 3
 * Nome progetto: IO -ver. c - Revisione: 2012-10-06
 * Autore: A. Giannico
 * Descrizione:
     Il programma è equivalente al precedente IO-vers. b e mostra un altro
     modo di referenziare il singolo bit di un registro di configurazione
     Invece di TRISx.y possiamo scrivere TRISxy_bit.
 *   MCU:PIC18F877; Sch. di sviluppo:Easypic3; Oscillatore: HS 8.0000 MHz Crystal
     SW: mikroPascal PRO for PIC  ver. 5.61
     http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
  }
program IO_c;
{ Dichiarazione variabili }
begin
  // configurazione della direzione dei singoli pin di I/O
  TRISA0_bit := 1;  // A0, A1, A2 vengono configurati come ingressi
  TRISA1_bit := 1;
  TRISA2_bit := 1;
  TRISB0_bit := 0;  // B0, B1 e B2 vengono configurati come uscite
  TRISB1_bit := 0;
  TRISB2_bit := 0;
  while TRUE do
    begin
     if (PORTA.0=1) then PORTB.0:=1  // equivalente a
                    else PORTB.0:=0; // if (PORTA.B0=1) then PORTB.B0:=1;
     if (PORTA.1=1) then PORTB.1:=1  //                   else PORTB.B0:=0;
                    else PORTB.1:=0;
     if (PORTA.2=1) then PORTB.2:=1
                    else PORTB.2:=0;
    end;       // Fine Main program
end.
```

*Figura 12: esempio n. 3 (IO-ver.c).*

Passando alle applicazioni pratiche, vedremo come sia semplice, mediante l'impiego di semplici pull-up, collegare contatti come quelli dei dip-switch al PIC, al fine di eseguire esperimenti simulando gli stati logici provenienti da altri circuiti come contatti di allarme o di diagnostica.

## LE PORTE DEL PIC UTILIZZATE COME INGRESSI DI STATI LOGICI

Come abbiamo avuto modo di sottolineare nella scorsa lezione, la maggior parte dei pin del PIC multiplexano più funzioni. In particolare, i pin di uscita possono essere utilizzati anche come ingressi, tanto che più in generale si parla di porte di I/O. Più precisamente, negli esempi applicativi che adesso mostreremo, utilizzeremo una parte dei pin del PIC come ingressi digitali e una parte come uscite digitali.
Se si vuole operare in tal senso è necessario configurare in firmware i relativi registri di direzione. Se si prende in considerazione, per esempio, la porta A, sarà necessario assegnare ai bit del registro di direzione TRISA il valore 1 se si vogliono impiegare come ingressi i pin corrispondenti. Questo aspetto apparirà chiaro nell'analisi diretta degli esempi che seguono.

## IL COSTRUTTO IF-THEN-ELSE

### Esempio n.1: IO-ver.a

Il programma mostra come pilotare determinati pin configurati come uscite in funzione degli stati logici presenti su altri pin configurati come ingressi. Si tratta del più semplice programma possibile, nel quale il flusso di programma, e con esso i valori delle uscite, venga a dipendere dal valore assunto da variabili di ingresso. Il flusso del programma non è quindi statico ma definito dinamicamente istante per istante in funzione del valore assunto dagli ingressi. Affinché ciò sia possibile è necessario che il programma contenga almeno l'esecuzione condizionata di una istruzione. Il costrutto MikroPascal più semplice che consente di ottenere un'esecuzione condizionata di istruzioni è il costrutto IF-THEN. Prima di entrare nel merito, osserviamo **(figura 6)** come in testa al codice vi sia la configurazione dei registri di direzione delle porte. In particolare, attraverso le istruzioni:

```
TRISA.0 := 1;
TRISA.1 := 1;
TRISA.2 := 1;
```

si configurano i pin A0, A1 e A2 come ingressi, mentre con le istruzioni:

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 3
 * Nome progetto: Lampeggio Led -ver. d - Revisione: 2012-10-08
 * Autore: A. Giannico
 * Descrizione:
     Il programma mostrare come sia possibile implementare un generatore
     di forma d'onda quadra con frequenza selezionabile tra 0,5 KHz (T=2ms),
     50Hz (T=20 ms) e 5 Hz (T=200 ms) attraverso dip-switch posti sugli
     ingressi B0 e B1. La forma d'onda viene raccolta sull'uscita A0 impiegando
     la funzione Delay_ms(). T_2 sta per T/2
 *   MCU:PIC18F877
     Scheda di sviluppo:Easypic3; Oscillatore: HS 8.0000 MHz Crystal
     SW: mikroPascal PRO for PIC  ver. 5.61
     http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
  }
program IO_d;
var set_T,T_2,i:byte;  { Dichiarazione variabili }
begin
  TRISB := 0xFF;       // Porta B configurata come porta di ingresso
  TRISA := 0x00;       // Porta A configurata come porta di uscita
  while TRUE do
    begin
     set_T:=(PORTB)and(%00000011);
     if (set_T=1) then T_2:=1
                  else if (set_T=2) then T_2:=10
                  else if (set_T=3) then T_2:=100;
     if (set_T>0) then
                  begin
                   PORTA.0:=1;   // *
                   for i := 1 to T_2 do
                             Delay_ms(1);// 1 millisecondo di ritardo
                   PORTA.0:=0;   // **
                   for i := 1 to T_2 do
                             Delay_ms(1);// 1 millisecondo di ritardo
                  end;
    end;
```

*Figura 13: esempio n. 4 (IO ver.d).*

Figura 14: schema elettrico - listato esempio ver.d e ver.f.

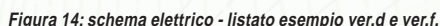
```
TRISB.0 := 0;
TRISB.1 := 0;
TRISB.2 := 0;
```

si configurano i pin B0, B1 e B2 come uscite.
Valgono le considerazioni fatte nella scorsa lezione per quanto riguarda l'indirizzamento del singolo bit del registro di configurazione TRISx della porta x e per quanto riguarda l'impiego delle notazioni decimale, binaria ed esadecimale.
Tornando al codice di **figura 6**, esso non fa altro che una assegnazione uno ad uno tra i valori assunti rispettivamente dagli ingressi A0, A1 e A2 e quelli delle uscite B0, B1 e B2. Da un punto di vista hardware, la simulazione degli ingressi è ottenuta attraverso dip-switch e resistori di pull-up **(figura 7)** mentre il valore delle uscite è reso evidente dallo stato assunto da altrettanti diodi led **(figura 7)**.
Come abbiamo fatto osservare, con i commenti inseriti nel codice, scrivere

```
if (PORTA.0=1) then PORTB.0:=1;
```

è del tutto equivalente a scrivere:

```
if (PORTA.B0=1) then PORTB.B0:=1;
```

In altre parole, la notazione PORTx.y è equivalente alla notazione PORTx.By. Da questo momento in poi utilizzeremo prevalentemente la prima delle due notazioni per due principali ragioni:

1. è una notazione più compatta;
2. consente di incorrere meno facilmente in errori.

Per esempio, PORTA.B0, contenendo sia la lettera A sia la lettera B può, per svista del programmatore, essere scambiato per PORTB.0 mentre scrivendo PORTA.0 la possibilità di commettere tale svista è quasi nulla.
L'elemento nuovo rispetto agli esempi mostrati nella scorsa lezione è aver utilizzato dei pin come ingressi, condizionando al valore assunto da questi il valore assegnato alle uscite. Quest'ultima operazione (assegnazione condizionata) è effettuata attraverso l'istruzione, o meglio il costrutto

```
if (condizione) then istruzione;
```

che significa letteralmente "Se (if appunto) la condizione in parentesi è verificata allora (then) esegui l'istruzione".
Nel caso in cui le istruzioni da eseguire, condizionatamente al verificarsi della stessa condizione, siano più di una, esse costituiscano un blocco, pertanto la precedente sintassi assume la forma di **figura 8**.
In questo caso l'intero blocco di istruzioni da eseguire codizionatamente al verificarsi della condizione è racchiusa tra la coppia di parole chiave Begin e end. Tale blocco assume pertanto il ruolo, per così dire, di macroistruzione.
Nell'esempio mostrato, la condizione è estremamente semplice ma in generale potrebbe assumere forma molto più complessa contenente diversi operatori logici, argomento di cui ci occuperemo in maniera dettagliata in lezioni successive. È tuttavia fin da ora evidente come l'istruzione condizionata da eseguire possa rappresentare l'uscita di una funzione combinatoria più o meno complessa di un certo numero di stati logici che fungono, in seno all'applicazione, da ingressi. Nel senso più generale del termine, infatti, il codice altro non è che l'implementazione in firmware di una macchina combinatoria nel caso in cui l'uscita o le uscite in ciascun istante dipendono solo dallo stato degli ingressi in quell'istante (macchina quindi senza memoria) oppure di una macchina sequenziale in cui l'uscita in un certo istante non dipende unicamente dal valore degli ingressi in quell'istante ma anche dallo stato della macchina in quell'istante, stato che dipende dalla storia pregressa della stessa che, di conseguenza, viene a essere, a tutti gli effetti, una macchina con memoria.

### Esempio n.2: IO-ver.b

In questo esempio **(figura 10)** mostriamo l'impiego del costrutto IF-THEN-ELSE, il quale altro non è che una generalizzazione, peraltro del tutto naturale e ovvia, del costrutto IF THEN, dal momento che prevede, oltre al ramo THEN, da eseguire in caso di verifica positiva della condizione, anche il ramo di flusso ELSE, da eseguire in caso di verifica negativa della condizione.
Il modo in cui abbiamo scritto le istruzioni condizionate nell'esempio ver.a **(figura 6)** è oltre che poco efficiente dal punto di vista della lunghezza del codice .hex anche poco elegante. Infatti, invece di utilizzare due istruzioni IF-THEN in sequenza, viste che esse servono in questo caso a contemplare le due uniche situazioni mutuamente esclusive, è possibile utilizzare il costrutto IF-THEN-ELSE **(figura 9)**.
La sintassi letteralmente significa "Se la condizione in parentesi è vera allora esegui l'istruzione 1, altrimenti esegui l'istruzione 2". Si noti il ";" dopo l'istruzione 2 a chiudere l'intera istruzione.
In effetti, l'istruzione IF-THEN-ELSE può essere scritta in maniera del tutto equivalente con due istruzioni IF-THEN in sequenza. Il confronto tra IO-ver.a **(figura 6)**

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 3
 * Nome progetto: Lampeggio Led -ver. e - Revisione: 2012-10-09
 * Autore: A. Giannico
 * Descrizione:
   Programma equivalenete ad IO ver. a, b, c. Il confronto, mostra in modo
   quasi banale quanto sia importante il modo in cui si esprime un algoritmo
   ai fini della compattezza e della leggibilità del codice. In questo caso,
   le tre istruzioni IF-THEN-ELSE erano infatti del tutto non necessarie.
 *  MCU:PIC18F877
   Scheda di sviluppo:Easypic3
   Oscillatore: HS 8.0000 MHz Crystal
   SW: mikroPascal PRO for PIC  ver. 5.61
   http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
 }
program IO_e;
{ Dichiarazione variabili }
begin
  // configurazione della direzione dei singoli pin di I/O
  TRISA.0 := 1;      // A0,A1. A2 vengono configurati come ingressi
  TRISA.1 := 1;
  TRISA.2 := 1;
  TRISB.0 := 0;      // B0,B1,B2 vengono configurati come uscite
  TRISB.1 := 0;
  TRISB.2 := 0;
  while TRUE do
    begin
     PORTB.0:=PORTA.0;  //equivalente a if ((PORTA.B0)=1) then PORTB.B0:=1;
     PORTB.1:=PORTA.1;  //                                 else PORTB.B0:=0;
     PORTB.2:=PORTA.2;
    end;             // Fine Main program
end.
```

*Figura 15: esempio n. 5 (IO ver. e).*

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 3
 * Nome progetto: Lampeggio Led -ver. f - Revisione: 2012-10-09
 * Autore: A. Giannico
 * Descrizione:
   Il programma è leggermente differente dalla precedente ver.d. Alla porta B
   (ingresso impostato tramite 8 dip-switch) è assegnato il significato di
   semiperiodo espresso in millisecondi. Tale valore diventa argomento della
   funzione di Vdelay_ms(t) che a differenza di Delay_ms() è in grado di
   accettare in ingresso una variabile consentendo in tal modo la generazione
   del ritardo utile all'ottenimento della forma d'onda quadra.
 *  MCU:PIC18F877; Scheda di sviluppo:Easypic3
   Oscillatore: HS 8.0000 MHz Crystal; SW: mikroPascal PRO for PIC  ver. 5.61
   http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
 }
program IO_f;
var T_2:byte;  { Dichiarazione variabili }

begin

  TRISB := 0xFF;       // Porta B: ingresso
  TRISA := 0x00;       // Porta A: uscita
  while TRUE do
    begin
     T_2:=PORTB;       // lettura del semiperiodo 0<T_2<255 ms
     Vdelay_ms(T_2);   // per mezzo semiperiodo uscita A0 a livello alto
     PORTA.0:=1;
     Vdelay_ms(T_2);   // per mezzo semiperiodo uscita A0 a livello basso.
     PORTA.0:=0;
    end;               // Fine Main Program
end.
```

*Figura 16: esempio n. 6 (IO ver.f).*

```
if (condizione_1) then
                   if  (condizione_2) then istruzione_1
                                      else istruzione_2;
```

*Figura 17: IF-THEN-ELSE nidificati.*

```
if (condizione_1) then
                   Begin
                    if  condizione_2 then istruzione_1
                                     else istruzione_2;
                   end;
```

*Figura 18: sintassi del tutto equivalente a quella di figura 17.*

```
if condizione_1 then
                 Begin
                  if  condizione_2 then istruzione_1;
                 end
                 else istruzione_2;
```

*Figura 19: si confronti la porzione di codice con quella di figura 18. Le due sintassi possono apparire simili ma hanno effetto completamente differente sul flusso di programma.*

e IO ver.B **(figura 10)** mostra questo interessante aspetto.
A basso livello, la condizione THEN corrisponde a un salto, unica istruzione, a basso livello appunto, in grado di far variare il flusso esecutivo di un programma. A basso livello non è infatti possibile esprimere la condizione ELSE nello stesso modo in cui è possibile fare ad alto livello ma è necessaria una seconda istruzione di salto condizionato.
D'ora in avanti, lavorando ad alto livello, tutte le volte che abbiamo due blocchi di istruzioni condizionate mutuamente esclusive utilizzeremo l'istruzione IF-THEN-ELSE piuttosto che due istruzioni IF-THEN.
È importante osservare come in questo caso istruzione_1 non termini con il ";" poiché essa fa parte dell'istruzione IF la quale non è ancora terminata ma continua con l'alternativa espressa dal ramo ELSE del costrutto. Il compilatore "conosce" e impiega questa regola di sintassi per decodificare l'istruzione essendo in questo modo in grado di riconoscere dove inizia e dove termina l'intero costrutto oltre che di isolare le singole parti che compongono l'istruzione, in particolare "condizione", "istruzione1" e "istruzione2".
La stessa sintassi vale nel caso in cui a essere condizionata al verificarsi della condizione non sia un'istruzione ma un gruppo di istruzioni, gruppo che come al solito racchiuderemo tra le parole chiave Begin e end **(figura 11)**.
Si noti l'assenza del ";" alla fine del primo "end" che è seguito dall'else. Il ramo else non viene infatti mai preceduto dal ";" dal momento che in quel punto l'istruzione non è ancora terminata. Il secondo "end" è invece dotato di ";" dal momento che con esso l'istruzione ha termine.

#### Esempio n.3:IO-ver.c

In questo esempio **(figura 12)** cogliamo semplicemente l'occasione per fare osservare come la notazione che consente di configurazione un bit di un registro di direzione

TRISx.y

dove x è la porta e y è il bit che si sta referenziando del registro, è del tutto equivalente a

TRISxy_bit

Per esempio, scrivere "TRISA.0:=1" è del tutto equivalente a scrivere "TRISA0_bit := 1". Nonostante tutto continueremo a impiegare la prima delle due notazioni essendo la più compatta e quindi immediata.

### CICLO FOR E PRIMO APPROCCIO ALLE VARIABILI

#### Esempio n.4: IO-ver.d

In questo esempio intendiamo utilizzare quanto imparato fino a questo momento per implementare un semplice generatore di forma d'onda rettangolare con frequenza impostabile. Osserviamo fin da ora che esistono modi più efficienti e corretti di generare forme d'onda quadre con un PIC, tuttavia operare come adesso vedremo è un primo modo, certamente semplice, di agire che ci consente di introdurre, in maniera altrettanto semplice, nuovi concetti.
Il periodo T viene impostato dall'operatore mediante due contatti di un dip-switch (Figura 13 e Figura 14). Alle quattro combinazioni possono evidentemente corrispondere quattro differenti frequenze di oscillazione che diventano 3 nel caso in cui una delle quattro combinazioni venga utilizzata per mettere in stand-by il circuito.
Il funzionamento del programma è abbastanza semplice. I pin di impostazione del periodo vengono letti dal controllore; in base alla configurazione riconosciuta viene definito il semiperiodo della forma d'onda espresso in millisecondi.
Più precisamente il controllore legge l'intera porta B come ingresso ma ne considera i soli 2 bit meno significativi grazie all'istruzione:

```
set_T:=(PORTB)and(%00000011);
```

che esegue l'AND bit a bit tra il contenuto della porta B e una maschera che consente di estrarre i soli due bit meno significativi. Questo tipo di operazione è piuttosto ricorrente in tutti i casi in cui si voglia estrarre da un byte specifici bit che occupano specifiche posizioni.
A questo punto, un ciclo FOR che esegue un numero di delay (ciascuno di durata un millisecondo) pari alla durata del semi-

periodo è sufficiente a protrarre rispettivamente prima un livello basso e quindi un livello alto della forma d'onda quadra. Terminato un ciclo, l'algoritmo si ripete. In firmware quindi un delay non serve semplicemente ad aspettare!
È facile comprendere come il programma possa facilmente essere generalizzato per un numero n di bit di ingresso aumentando il numero di possibili combinazioni e quindi di frequenze impostabili per la forma d'onda in uscita. In realtà, appare anche chiaro come il programma diventi difficilmente gestibile con semplici istruzioni IF-THEN-ELSE se si cominciano ad utilizzare, per esempio, otto bit di ingresso che implicano ben 255 combinazioni. Evidentemente può essere opportuno stilare il codice in maniera più efficiente per renderlo compatto. Un modo di ottenere ciò sarà mostrato nell'ultimo esempio di questa lezione, mentre un secondo modo di compattare il codice potrebbe essere quello di impiegare il costrutto CASE di cui avremo modo di parlare nelle prossime lezioni. Completiamo la discussione del codice IO-ver.d con alcune semplici ma utili osservazioni:
1) se si desidera che dopo un certo periodo di funzionamento il firmware venga messo in condizione di stand-by con uscita a livello basso impostando la combinazione B1B0=00, allora è necessario che le due istruzioni indicate con (*) e (**) nel listato di **figura 13** si trovino nella posizione mostrata nella stessa figura. In caso opposto, cioè scambiando le istruzioni (*) e (**), alla condizione di stand-by corrisponde uscita fissa a livello alto. La questione è importante se si pensa che questo segnale potrebbe pilotare qualcos'altro e che il pilotaggio potrebbe avvenire sul livello piuttosto che sulla transizione.
2) è necessario distinguere bene il simbolo di uguale (=) da quello di assegnazione (:=). Il primo può essere utilizzato per esprimere condizioni mentre il secondo in Mikropascal viene utilizzato come operatore di assegnazione di un valore a una variabile;
3) è necessario porre attenzione alla corrispondenza tra range di variazione di una variabile (in base al tipo) e range effettivo di impiego. Nel nostro caso è stata utilizzata una sola variabile, precisamente di tipo byte, per cui il valore intero massimo che essa può assumere è 255.
4) è necessario fare attenzione al fatto che MikroPascal è un compilatore non case sensitive. Questo in generale potrebbe apparire un vantaggio, tuttavia non sempre lo è. Per esempio, nel listato IO-d, se al posto di set_T avessimo T e ponessimo t al posto di i senza dichiararlo, il compilatore non segnalerebbe alcun errore poiché per esso t e T sono la medesima cosa. Il programma tuttavia funzionerebbe in maniera del tutto errata.
Non è difficile modificare il programma in maniera tale da poter impostare oltre che la frequenza dell'onda anche il duty-cicle. Lasciamo al lettore che vuole cimentarsi in questa operazione questo semplice esercizio.

## PRIMI CENNI ALL'OTTIMIZZAZIONE DEL CODICE

### Esempio n.5: IO-ver.e

Questo listato di codice non è altro che una variante dei listati IO-versioni a, b, e c. La funzione implementata può infatti essere ottenuta assegnando direttamente alle uscite il valore dei corrispondenti ingressi. Questo esempio del tutto elementare ci fa comprendere come una stessa funzione possa spesso essere implementata in differenti modi. Un buon programmatore non deve mai trascurare questo aspetto per due principali motivi:
1) un modo di scrivere un flusso di programma piuttosto che un altro potrebbe rendere lo stesso più o meno leggibile;
2) in fase di compilazione si possono ottenere codici .hex di lunghezza differente anche se gli stessi eseguono nel complesso la stessa funzione. In altre parole, il codice può essere più o meno ottimizzato dal punto di vista della compilazione e della quantità di memoria flash occupata.
Questo suggerisce una prassi sempre opportuna nella programmazione ovvero quella di riesaminare il codice dopo averlo scritto, anche se perfettamente funzionante, al fine di individuarne affinamenti che possano renderlo più compatto, più leggibile e più leggero dunque, in altre parole, più ottimizzato.

### Esempio n.6: IO-ver.f

Abbiamo detto che quando si scrive un codice è sempre necessario chiedersi se il modo in cui lo si sta scrivendo sia quello più efficiente e leggibile. L'esempio, per quanto banale, di **figura 15** confrontato con quelli delle **figure 6**, **10** e **12** è sufficiente a chiarire il concetto. Se si vuole generalizzare il codice di programma nella forma riportata nell'esempio della versione d facendo semplicemente uso dell'istruzione IF-THEN ELSE per contemplare tutte le possibili combinazioni di 8 bit di ingresso, il codice rischia di divenire ben presto quasi ingestibile.
Ottimizzazioni come quelle che comportano l'impiego del costrutto CASE o ancor meglio una diversa logica di impostazione del periodo della forma d'onda, come mostrato nell'esempio della versione f **(figura 16)**, consentono spesso miglioramenti funzionali, in termini di compattezza, oltre che una maggiore leggibilità. È importante notare, inoltre, che fino a

quando la frequenza di clock non risulta particolarmente elevata, il tempo impiegato per l'esecuzione delle istruzioni all'interno del ciclo è trascurabile rispetto a quello di delay. Nel caso invece in cui utilizzassimo la funzione delay_us per generare forme d'onda di frequenza prossima al MHz, questo non sarebbe più vero e l'errore che avremmo sulla forma d'onda generata non sarebbe più impercettibile e quindi trascurabile. Nelle puntate che seguiranno torneremo a prendere in esame questa interessante applicazione al fine di mostrare come poter aggirare questo problema.
Ovviamente, una funzione del tipo implementata è puramente combinatoria nel senso che il valore dell'uscita in un certo istante dipende esclusivamente dal valore degli ingressi in quello stesso istante, come avviene per i circuiti digitali combinatori. Il vantaggio di utilizzare un controllore per l'implementazione di una funzione combinatoria piuttosto che una logica digitale puramente hardware sta nel fatto di disporre di una logica che, fissati gli ingressi e le uscite, può essere riprogrammata senza modificare l'hardware. Lo svantaggio sta invece nel fatto che il controllore necessita di programmazione. Un altro piccolo svantaggio può aversi in termini di velocità di riposta dal momento che un circuito puramente digitale essendo composto da sola logica hardware potrà in genere raggiungere velocità di risposta superiori.

### Istruzioni IF-THEN-ELSE nidificate

È opportuno spendere alcune parole sull'istruzione F-THEN ELSE nidificata. Si tratta di istruzioni IF-THEN-ELSE che dipendono esse stesse da una condizione IF-THEN. Nel caso in cui il livello di nidificazione sia unico abbiamo un costrutto del tipo mostrato in **figura 17**. Ancora una volta al posto di istruzione_1 ed istruzione_2 potremmo avere gruppi o blocchi di istruzioni racchiusi ciascuno in una coppia di parole riservate Begin e end.
Lo stralcio di codice di **figura 17** potrebbe creare qualche dubbio di interpretazione del flusso. In particolare si potrebbero avere dubbi nell'associare istruzione_2 al secondo ramo else piuttosto che al primo. Al fine di fugare qualunque indecisione è opportuno segnalare che il compilatore interpreta la precedente sintassi di **figura 17** in maniera del tutto equivalente a quella di **figura 18**, per cui l'else risulta, in ultima analisi, associato al secondo if.
Se vogliamo invece associare l'else al primo if è necessario scrivere il costrutto nella forma riportata in **figura 19**.
Come è ovvio comprendere, le due sintassi imprimono al flusso di programma un'evoluzione completamente differente. Appare a questo punto ovvio come il discorso sui livelli di nidificazione delle istruzioni IF-THEN-ELSE possa essere esteso a un numero n generico di livelli.

### Le funzioni Delay_us, Delay_ms e Vdelay_ms

Abbiamo già discusso dei problemi che potremmo avere in termini di errore relativo della forma d'onda nel caso in cui si impieghino i listati mostrati fino a questo punto per ottenere frequenze di clock vicine al megahertz sfruttando la funzione Delay_us().
Qualche osservazione va invece fatta in merito all'argomento delle funzioni Delay_us() e Delay_ms(), argomento che deve essere un valore e non una variabile. Nel caso in cui si voglia utilizzare come argomento una variabile è necessario infatti utilizzare Vdelay_ms() piuttosto che Delay_ms(). Di seguito riassumiamo brevemente quanto è necessario sapere sulle funzioni Delay_ms(), Delay_us() e VDelay_ms().

#### Delay_us

Si tratta di una Procedure la cui esecuzione non ritorna alcun dato. Essa serve a creare un ritardo in microsecondi pari al valore che viene a essa passato come argomento. L'argomento deve necessariamente essere un valore numerico definito cioè una costante; non può cioè trattarsi di una variabile. Per esempio,

```
Delay_us(1000);
```

sviluppa un ritardo di 1.000 microsecondi cioè 1 millisecondo.
Essa accetta come valori in ingresso delle word, cioè dati a 16 bit il cui range può per questo essere compreso tra 0 e 65.535. Evidentemente, se si vogliono ottenere ritardi maggiori è necessario iterare l'istruzione oppure utilizzare la procedura Delay_ms().

#### Delay_ms()

Valgono considerazioni analoghe a quelle fatte per Delay_us con l'unica differenza che l'argomento esprime la durata del ritardo espresso in millisecondi. Per esempio:

```
Delay_ms(1000);
```

sviluppa un ritardo di 1.000 mllisecondi cioè un secondo. I valori in ingresso sono delle word, cioè dati a 16 bit il cui range può per questo essere compreso tra 0 e 65.535 (due elevato a 16 meno 1). Evidentemente, se si vogliono ottenere ritardi maggiori è necessario iterare la stessa istruzione un numero sufficiente di volte.

#### Vdelay_ms()

Consente di creare un ritardo espresso in millisecondi pari al valore assunto da una variabile piuttosto che da una costante. Per esempio,

```
pause:=1000;
Vdelay_ms(pause);
```

consente di ottenere un ritardo di 1.000 millisecondi, cioè un secondo. Le prima sostanziale differenza funzionale rispetto a Delay_ms sta nel fatto che il valore del ritardo non è noto a priori ma viene determinato a runtime e corrisponde al valore assunto dalla variabile. La seconda sta nel fatto che la variabile oggetto della funzione è in generale una dword cioè double word, ovvero una variabile a 16 bit da cui scaturisce un range di variabilità compreso tra 0 e 4.294.967.295 (cioè 2 elevato a 32 meno 1).

### La definizione delle variabili

Abbiamo avuto già modo di accennare al fatto che il MikroPascal, come tutti i compilatori Pascal, consente un controllo abbastanza forte sul tipo delle variabili. Nell'esempio ver.f abbiamo impiegato per la prima volta una variabile all'interno di un codice. La definizione di una variabile globale avviene sempre dopo la parola riservata Program e prima della parola riservata Begin, secondo la seguente sintassi:

## PER approfondire...


- [1] PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers (www.mikrochip.com);
- [2] Sito di Microchip (www.mikrochip.com);
- [3] MikroPascal PRO for PIC v.5.61 demo e documentazione relativa (www.mikroe.com);
- [4] Help in linea MikroPascal Pro for PIC v. 5.61.


```
var nome_variabile:Tipo_variabile;
```

In questo caso abbiamo utilizzato un avariabile di tipo byte legata al fatto che la porta da cui leggiamo il valore da assegnare alla variabile T_2 è appunto un gruppo di 8 bit componenti PORT_B **(figura 14)**. Nel prosieguo del corso torneremo più volte ad approfondire l'argomento e a fare osservazioni sulla dichiarazione delle variabili poiché un loro corretto impiego è fondamentale non solo dal punto di vista funzionale del codice ma anche dal punto di vista della sua compattezza e della sua ottimizzazione.
In generale, per variabile si intende qualcosa il cui valore può variare durante l'esecuzione del codice, cioè a runtime. La variabile viene dichiarata una sola volta. Questo significa che un nome identificativo può essere utilizzato per identificare una sola variabile. A basso livello, dal punto di vista del compilatore, la variabile altro non è che una porzione di memoria riservata per contenerne il valore. Questo spiega perché il suo tipo deve essere dichiarato e perché la sua dichiarazione deve sempre precedere il suo impiego. Questo è il motivo per cui una variabile globale viene sempre dichiarata prima della parola riservata Begin che dà inizio al blocco di codice, secondo la sintassi:

```
var lista_di_identificatori : tipo;
```

dove lista_di_identificatori altro non è che la lista degli identificatori delle variabili separate tra loro da una virgola. Questo consente di evitare di dichiarare separatamente ciascuna variabile e di accomunare piuttosto tutte le variabili di uno stesso tipo sotto un'unica dichiarazione, compattando il codice e incrementandone la leggibilità. Per esempio, secondo tale logica potremo scrivere:

```
var i, j, k : byte;
counter, temp : word;
samples : array[100] of word;
```

A ciascuna variabile dichiarata corrisponde una porzione di memoria RAM assegnata e riservata dal compilatore. Per questo, non solo il numero di variabili dichiarate ma anche il tipo di ciascuna di esse contribuisce a determinare la quantità di RAM richiesta dall'applicazione. Ciò è in realtà vero soprattutto per le variabili globali poiché il compilatore cerca di riutilizzare, per quanto possibile, lo spazio di memoria richiesto da variabili locali dichiarate in funzioni e procedure differenti. L'argomento risulterà più chiaro quando avremo modo di trattare la modularità del codice e quindi la scomposizione dello stesso in funzioni e procedure.

### OSSERVAZIONI E CONCLUSIONI

In conclusione, in questa seconda puntata dedicata alla programmazione dei controllori PIC in MikroPascal, abbiamo:

- imparato a utilizzare il costrutto IF-THEN-ELSE;
- imparato a utilizzare il costrutto FOR;
- preso confidenza con la dichiarazione delle variabili;
- imparato a utilizzare le porte del PIC come ingressi di stati logici;
- imparato a utilizzare le funzioni Delay_ms(), Delay_us() e Vdelay_ms().

Nelle prossime puntate continueremo a esplorare il linguaggio attraverso esempi applicativi; al momento opportuno non mancheremo di fare alcune osservazioni, anche di carattere hardware sulle applicazioni, utili alla realizzazione pratica dei nostri primi circuiti a microcontrollore. ❏

**CODICE MIP 2830209**

## Interfacciamento dei processori *(parte dodicesima)*

# PILOTARE I DISPLAY *a matrice di punti*

**La visualizzazione dell'informazione è solitamente affidata a display con due o più digit del tipo "a sette segmenti", ma sono sempre più diffusi i visualizzatori che si affidano alle matrici di LED, consentendo la magia dei testi scorrevoli e delle immagini animate; di questi componenti ci occuperemo in questa puntata.**

Dopo aver trattato l'interfacciamento dei visualizzatori a LED realizzati con digit "a sette segmenti" è doveroso occuparci anche dei display a matrice di punti. Essi sono molto diffusi e il loro utilizzo è decisamente accattivante, sia per il fatto di consentire la proposta di testi (statici o "scorrevoli") sia per la possibilità di gestire (pur con qualche limite di risoluzione) anche figure grafiche, immagini, esse pure in modo statico o animato; sono disponibili in batterie di LED (Dot Matrix LED Displays) di solito organizzate in 5 Colonne e 7 Righe (5x7) oppure in matrici 8x8, più adatte alla gestione delle immagini, specialmente se ordinate in pannelli di grandi dimensioni, unendole tra loro da ogni lato per realizzare superfici omogenee e regolari, eccellenti per fornire informazioni, specialmente nel campo della pubblicità o, per esempio, sulle nostre strade, per le comunicazioni sulla viabilità o di pubblica utilità.
Sebbene (nel contesto che ci siamo prefissati di analizzare) la cosa sia marginale, va detto che gli stessi pannelli possono essere realizzati anche con LED discreti (sempre organizzati a matrice), molto frequentemente di tipo RGB (acronimo di Red, Green, Blue), così da poter disporre di tutti i colori resi possibili dalla miscela di queste tre tinte fondamentali: è fuori discussione che la gestione di ciascun singolo dot pixel sarà molto più complessa di quella necessaria nei visualizzatori monocromatici, limitata alla sola accensione o spegnimento del LED ad esso associato.
Il mercato offre una vasta gamma di modelli per questa periferica optoelettronica, diversi per dimensione e/o per colore dei LED; di solito il progettista si limita a sce-

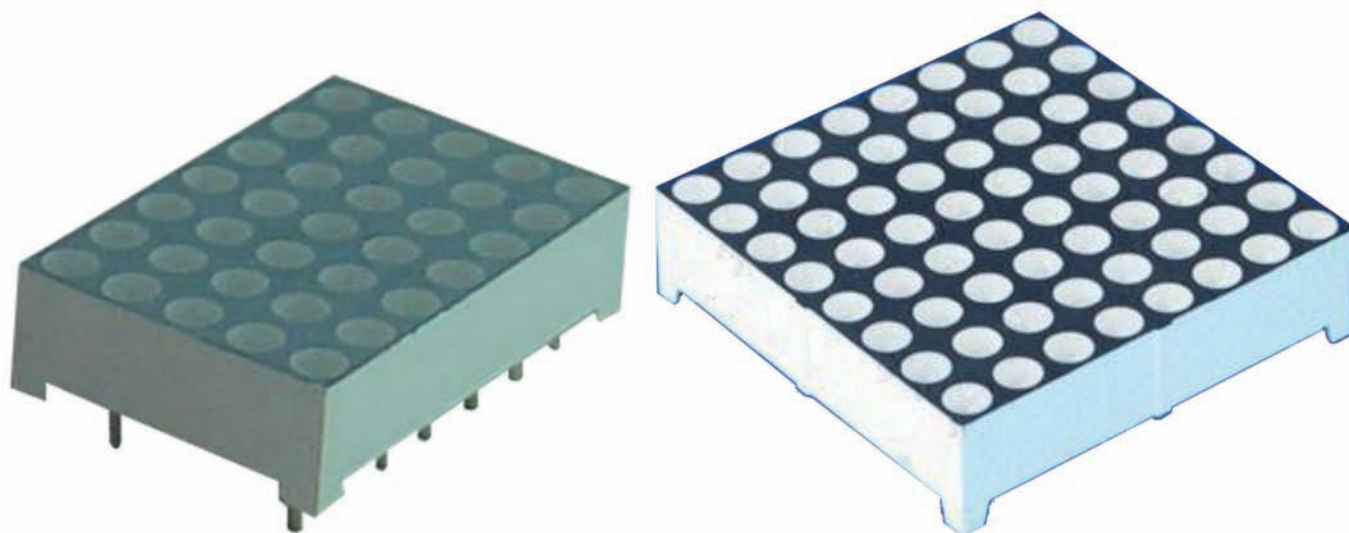



*Figura 1: alcuni modelli di Dot Matrix LED Display: INBM7571 (5x7) e HS-2088 (8x8).*

di GIORGIO OBER

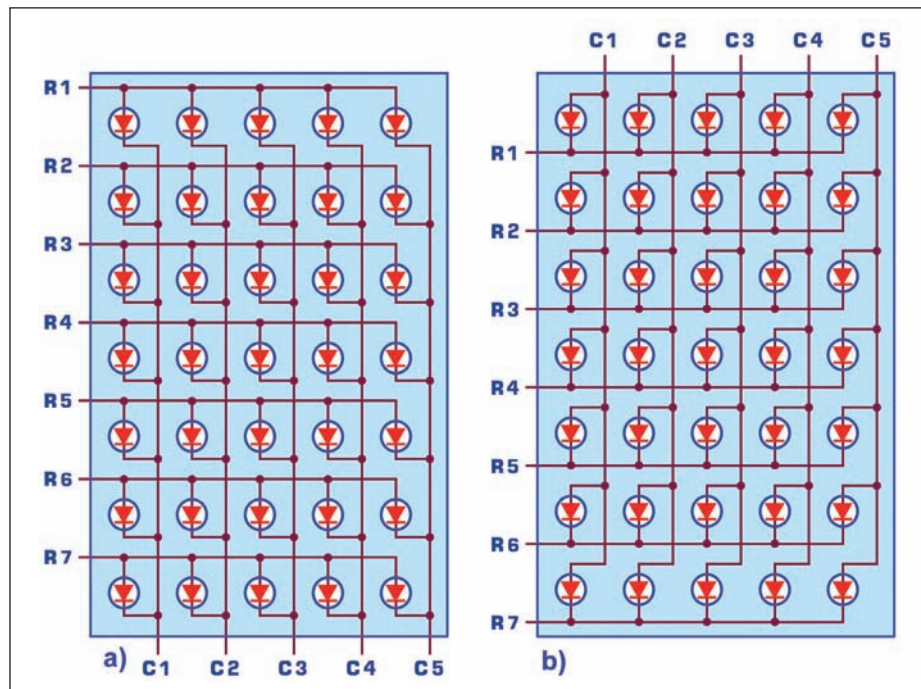



Figura 2: Dot Matrix 5x7 LED Display: a) a catodo comune; b) ad anodo comune.

gliere in funzione della sua forma estetica o delle proprie necessità d'ingombro, non curandosi (come invece succede per altri componenti) di conoscere il suo produttore: per questa ragione le sigle dei Display sono sostanzialmente irrilevanti. La **figura 1** ne mostra un paio, scelti nella forma tipica delle due matrici 5x7 e 8x8.

La tecnologia coinvolta nella matrice monocromatica è quella dei LED che la compongono: arseniuro di gallio e alluminio (AlGaAs, per i rossi), nitruro di gallio e indio (InGaN, per i verdi), fosfuro di alluminio, gallio e indio (AlGaInP, per i gialli) o il seleniuro di zinco (ZnSe) e ancora InGaN, per i moderni blu e i bianchi). La tensione di lavoro ottimale (con corrente diretta di 20 mA), per ciascuno dei 35 o 64 LED della matrice, può essere ritenuta pari a: 1,8 V (rossi), 1,9 V (gialli), 2 V (verdi), 3 V (bianchi) e 3,5 V (blu).

Sembra dunque superfluo perdere tempo in considerazioni di pilotaggio, del tutto identiche a quelle dei normali LED, ma è evidente che la struttura di questi dispositivi è stata studiata proprio per ridurre al minimo il numero di piedini necessari (per questo scopo) sul suo contenitore; volendo gestire una matrice di 5x7 LED sarebbero necessari 35 pin (uno per ogni anodo), supposta la presenza di un trentaseiesimo che ne raccolga tutti i catodi, e viceversa, nel caso di matrici ad anodo comune.

L'idea è invece quella di dividere i LED in gruppi composti solo da una piccola parte di essi, collegandone insieme gli anodi in modo da costituire una "Riga", e di collegare poi tra loro i catodi dei LED che occupano la medesima posizione in ciascuna Riga, dando luogo a un numero di "Colonne" pari a quello dei Led in essa contenuti.

La **figura 2** mostra le due tipologie possibili, relative a matrici da 5 x 7 = 35 LED; anche in questo caso si può parlare di Dot Matrix a catodo o ad anodo comune, sebbene l'affermazione sia da ritenersi vera solo dal punto di vista delle Colonne LED.

In questo modo ogni LED ha le sue coordinate univoche, Rx,Cy (con x da 1 a 7 e y da 1 a 5), e potrà essere acceso, se a catodo comune, portando alta la Riga e bassa la Colonna a cui appartiene e viceversa, se ad anodo comune.

La presenza di due gruppi distinti di bit (uno da 7 e uno da 5) suggerisce la necessità di almeno due registri, a cui affidarli: l'utilizzo di un microcontrollore sembra fatto apposta per la gestione diretta di un display a matrice di punti; alle sue porte PortA e PortB possono essere affidate parole a 8 bit, lasciando intendere un facile controllo anche delle versioni 8x8 dot; il limitato numero di linee d'uscita di una porta parallela SPP di un Personal Computer non le consente, purtroppo, di fare altrettanto, ma vedremo come aggirare questo presunto insuperabile ostacolo.

In ogni caso è necessario ricorrere alla tec-

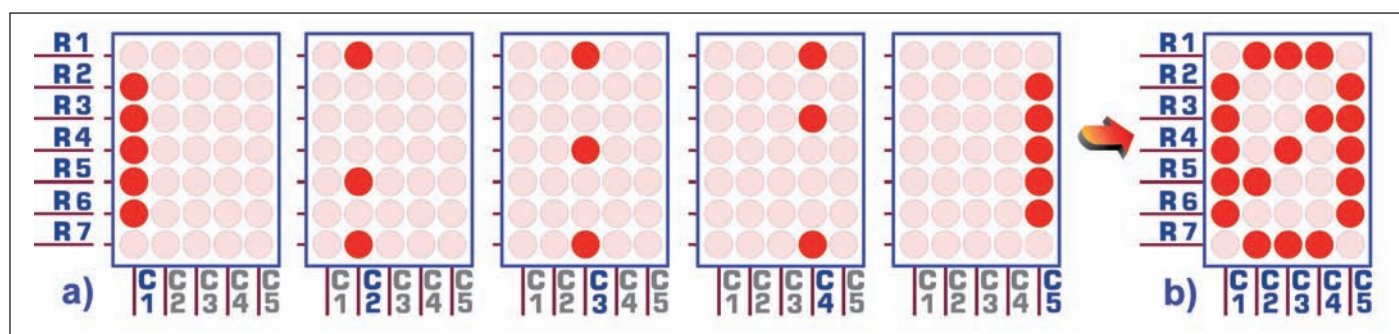



Figura 3: Dot Matrix 5x7 LED Display: sequenza da imporre per visualizzare il carattere "0".

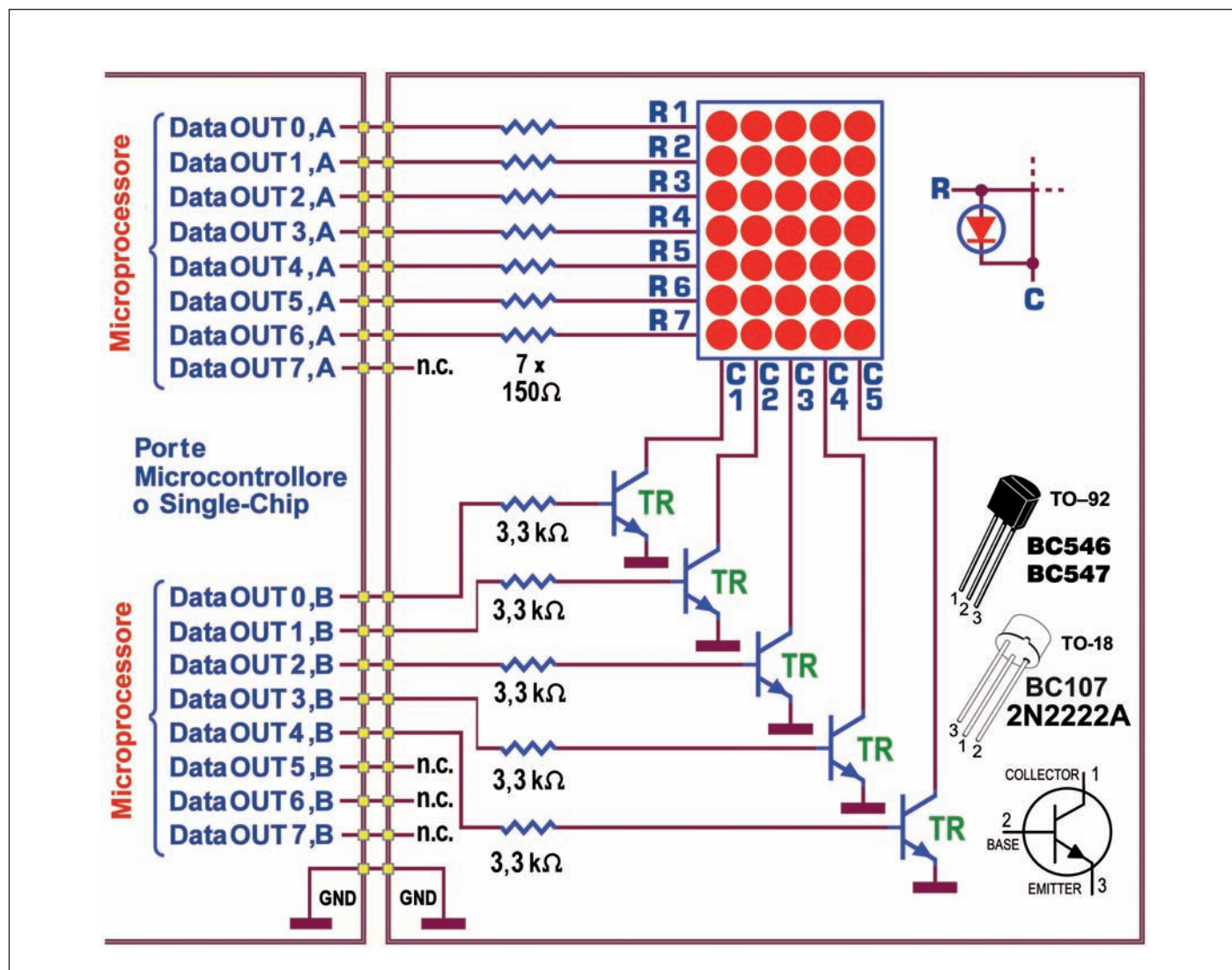
nica del multiplexing dell'informazione: i 35 bit richiesti non possono essere forniti simultaneamente ma devono essere frazionati in cinque parti (una per colonna) affidando in sequenza ciascuna di esse alla porta d'uscita associata alle sette Righe e attivando (cioè portando a massa) di volta in volta una sola Colonna.
Supponiamo di voler visualizzare il carattere numerico "0"; per poterlo fare è necessario anzitutto selezionare la Colonna C1 (portandola a massa) e lasciare inattive (alte) le altre, da C2 a C5; l'operazione successiva dovrà porre un'informazione sulle 7 Righe, così da influenzare solo i 7 LED connessi alla prima Colonna: nel nostro esempio l'aspetto della prima Colonna del carattere desiderato richiede di porre a 1 le righe da R2 a R6, lasciando a 0 la prima (R1) e l'ultima (R7). Successivamente, dopo aver attivato la sola colonna C2 (lasciando deselezionate tutte le altre), delle 7 Righe verranno attivate solo la prima (R1), la quinta (R5) e l'ultima (R7), e così via.
La **figura 3a** mostra l'aspetto della matrice di LED nelle cinque fasi richieste: è evidente che la sequenza non lascia intendere la presenza del carattere desiderato ma, se viene accelerata, esso comincerà a formarsi in modo sempre più nitido al crescere della frequenza di scansione, fino a risultare del tutto leggibile e privo di sfarfallamenti, non appena tale frequenza avrà raggiunto il così detto valore di fusione (Flicker Fusion), intorno ai 50 Hz.
Mentre scrivo mi torna alla mente un librettino "magico" di quando ero bambino che conteneva una serie di immagini leggermente diverse tra loro, da una pagina all'altra: quando le pagine (con abile movimento delle dita) venivano fatte scorrere in fretta, le immagini si animavano per incanto, lasciandomi a bocca aperta! Molto più tardi ho saputo che il "folioscopio" (o flipbook) risale alla metà dell'ottocento ed è stata una forma primitiva di animazione.
In entrambi i casi l'effetto è dovuto alla persistenza ottica, cioè alla capacità della retina dell'occhio umano di trattenere l'immagine per qualche frazione di secondo anche dopo che l'immagine stessa non è più visibile; la scansione rapida delle Colonne (maggiore di 50 volte al secondo), ciascuna con accesi solo i LED previsti dalle rispettive Righe, consente di



*Figura 4: Dot Matrix 5x7 LED Display: controllo con microcontrollore, schema 1.*

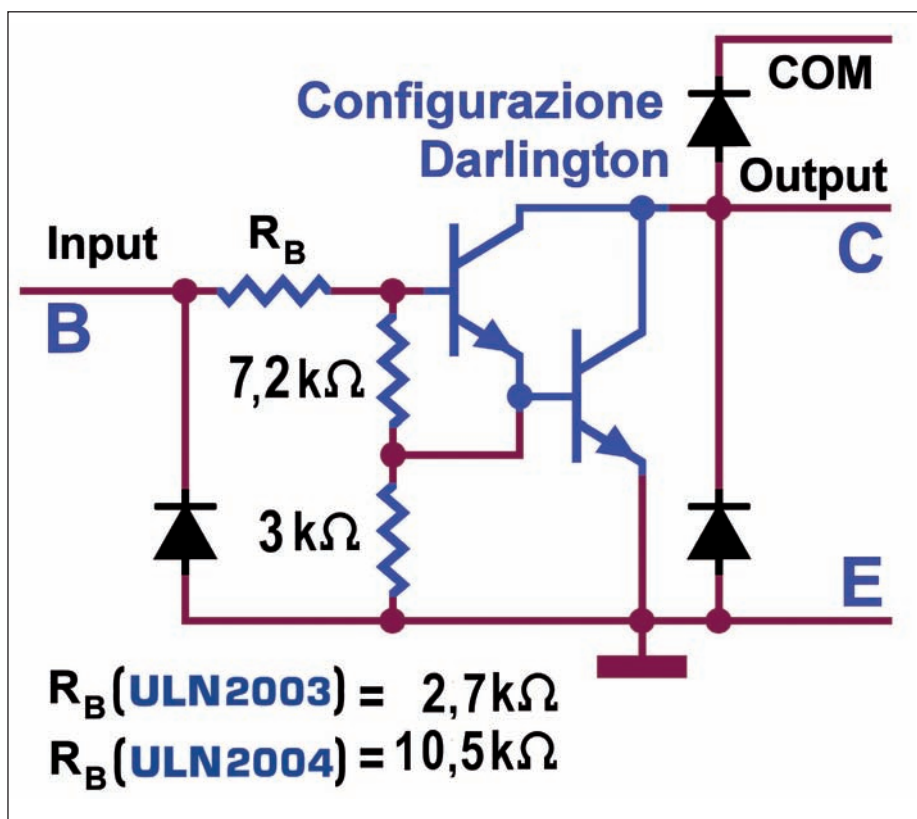



*Figura 5: High-voltage/high-current Darlington transistor arrays ULN2003: schema interno.*

percepire l'immagine completa del carattere, come si vede in **figura 3b**.
Naturalmente la cosa funziona perfettamente anche se l'informazione è fornita sulle Colonne e la scansione è imposta sulle Righe, ma è consuetudine utilizzare la tecnica descritta dalla **figura 3**; la proposta più naturale è dunque quella di affidare l'informazione ai primi 7 bit di una porta d'uscita e la gestione della scansione ai 5 bit di una seconda porta d'uscita; i bit rimanenti di ciascuna porta del microcontrollore possono ovviamente essere programmati per altri servizi, magari in ingresso, per leggere dei tasti in grado di organizzare la partenza e lo stop del processo di visualizzazione (con il bit7 di PortA) e il tipo di sequenza desiderata (con i bit5, bit6 e bit7 di PortB).
Per fissare i dettagli del progetto rimane ancora da analizzare la situazione delle correnti coinvolte:
a) di certo ogni Riga sarà chiamata a erogare la corrente di un solo LED, per cui possiamo ritenere che le linee d'uscita della porta coinvolta siano in grado di fornirla senza problemi: sarà opportuna dunque solo la presenza di un resistore in serie, per limitarne il valore; per eccesso di sicurezza potrebbe essere impiegato un 3-State Octal Buffers/Line Drivers non invertente 74LS244 tra le linee della porta e quelle di Riga della matrice;
b) per contro, ogni Colonna può essere costretta a gestire l'accensione di più LED, anche tutti e sette: nessuna logica programmabile dispone di linee in grado di assorbire la corrente necessaria, per cui si rende obbligatoria la presenza di uno strato di bufferizzazione.
La **figura 4** mostra la più immediata delle soluzioni con corrente di Colonna amministrata da semplici transistor bipolari general purpose di tipo NPN: la scelta non è critica poiché le condizioni di lavoro non richiedono tensioni applicate (Vcemax) elevate, né frequenze di lavoro critiche; molto comune è l'utilizzo del 2N2222A, apprezzabile per la discreta corrente di collettore (Ic=800mA; Vce=40V), ma vanno bene anche transistor come BC107 o BC547, anche se garantiscono entrambi corrente di collettore decisamente inferiore (Ic=100mA, con 200mA di picco; Vce=45V) ma comunque sufficiente per servire (nella condizione peggiore) l'accensione di 7 Led.
Da notare che il programma di gestione dovrà forzare a 1 le linee della PortB dato che il transistor buffer di corrente è di per sé un inverter: solo così la Colonna chiamata in causa verrà messa a massa, per un corretto servizio dei LED a catodo comune a essa associati.
L'utilizzo di transistor può essere favorito da parte dei "vecchi" appassionati, con cassetti pieni di componenti di recupero, ma la corrente necessaria per accendere con soddisfazione i LED di ciascuna colonna può essere assicurata anche con componenti specificatamente studiati per questo scopo; quelli più utilizzati (per la loro praticità e il loro costo contenuto) sono certamente gli High-voltage/High-current Darlington Transistor Arrays, disponibili in diverse versioni.
La caratteristica comune a tutti questi

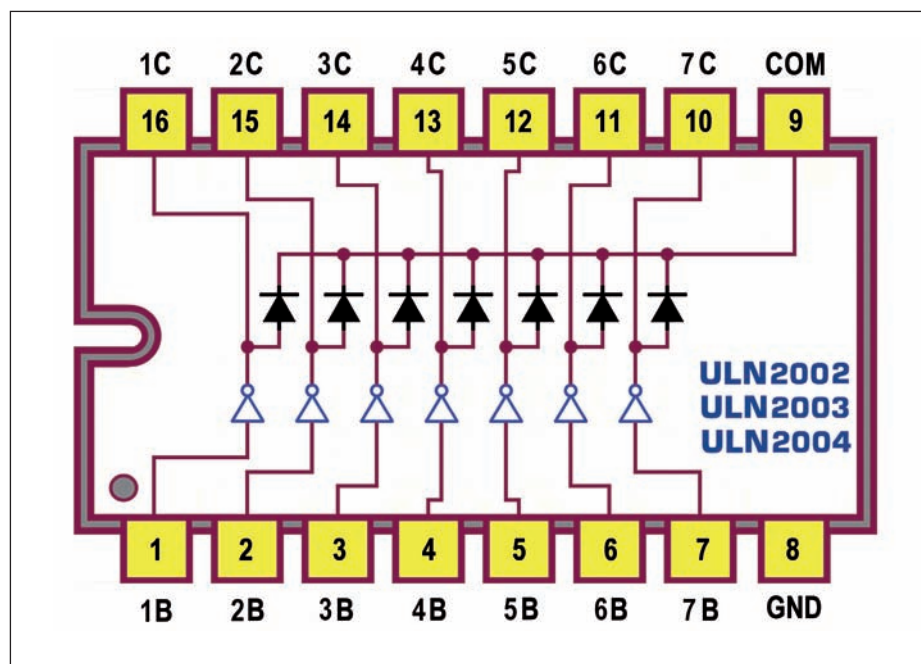



Figura 6: High-voltage/high-current Darlington transistor arrays ULN2003: pinout.

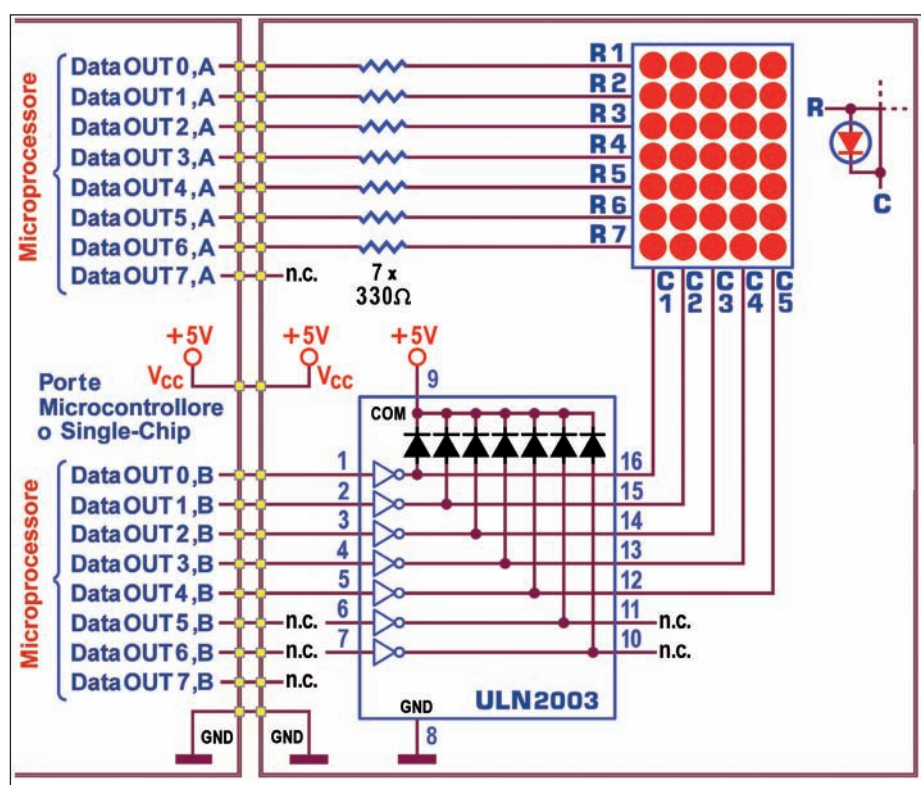



Figura 7: Dot Matrix 5x7 LED Display: controllo con microcontrollore, schema 2.

dispositivi è quella di utilizzare (tra l'ingresso e l'uscita di ciascun canale disponibile) una coppia di transistor in configurazione Darlington che, come è noto, si comporta come un singolo transistor ad alto guadagno di corrente; in questo modo la corrente gestita da ogni linea può arrivare fino a 500 mA, con picchi di 600 mA, consentendo anche alle normali logiche, TTL (come i sigle-chip o la porta Parallela di un PC) o CMOS, di pilotare periferiche che richiedono alta corrente.
Altra caratteristica comune a ogni modello di Darlington Transistor Arrays è quella di limitare la corrente d'ingresso con un resistore Rb in serie e di assicurare un buon grado di sicurezza con diodi polarizzati inversamente (clamp diodes), sia in ingresso sia in uscita; le uscite di ciascun canale sono a collettore aperto e consentono di utilizzare per la periferica a esse collegata tensioni di alimentazione diverse, anche maggiori, fino ad un massimo di 50 V, rispetto a quella (di tipo logico) usata per il controllo in ingresso, di solito di 4,2 V fino a un massimo di 5 V.
In aggiunta le stesse uscite possono essere facoltativamente collegate all'alimentazione della periferica con un ulteriore diodo di soppressione, previsto internamente, per dissipare eventuali possibili picchi di tensione durante il controllo di carichi induttivi; per motivi di logica praticità i catodi di tutti questi diodi opzionali interni sono collegati tra loro e resi disponibili su un piedino dedicato, detto COM.
Queste nostre considerazioni sono riassunte nello schema di **figura 5**, relative a ciascuno dei sette canali di un ULN2003 che, come detto, sono dei potenti driver d'uscita (high current sink drivers) capaci di assorbire fino a Ic=500mA, con Vcemax=50V. La **Figura 6** mostra il suo pinout.
Coerentemente si nota la necessità di un solo collegamento di alimentazione: la massa comune, sul pin 8; il ULN2003 è specificatamente pensato per l'utilizzo con logiche TTL (o per dispositivi CMOS alimentati a 5 V); in particolare, per questa ragione, il resistore di Base Rb (visibile in **Figura 5**) il valore di 2,7 kohm.
Esiste però la versione adatta ai dispositivi CMOS/PMOS alimentati da 6 V a 15 V: si tratta del ULN2004, funzionalmente identico e perfettamente pinout compatibile con il precedente; da notare che il resistore di Base Rb ha ora valore pari a 10,5 kohm.
I costruttori offrono altre alternative alla coppia appena descritta, tutti con pinout uguale a quello di **Figura 6**:
a) con capacità di assorbire corrente maggiore, Ic=600mA (e con Vce=50V): il ULN2012 (per uso con TTL) e il ULN2013 (per uso con CMOS);

b) con corrente tipica (Ic=500mA) ma con tensioni di lavoro molto più grandi: Vce=95V (il ULN2022 per TTL e il ULN2023 per CMOS) e Vce=100V (il 75468 per TTL e 75469 per CMOS).
La **Figura 7** mostra la soluzione che controlla la corrente di Colonna con questi dispositivi e, precisamente, data la tipologia delle logica programmabile utilizzata, con un ULN2003.
A margine di questa trattazione va detto che esiste anche la possibilità di disporre di componenti con otto transistor NPN Darlington indipendenti, ideali per pilotare matrici di 8x8 Led; la struttura funzionale e lo schema interno (**Figura 5**) sono gli stessi di quella descritti per i componenti con 7 canali e, anche in questo caso, viene reso disponibile un componente progettato per essere compatibile con lo standard famiglie TTL (il ULN2803) e uno (il ULN2804) ottimizzato per tollerare le maggiori tensioni (da 6V a 15V) in uso con i CMOS o i PMOS.
La **figura 8** mostra il pinout (valido per entrambi), comunque caratterizzato da Ic=500mA, Vce=50V. Esiste anche una coppia di componenti (ULN2981 per TTL e ULN2982 per CMOS, vedi **figura 9**) con 8 canali non invertenti sempre in grado di erogare 500 mA (con Vcemax=50V), pinout compatibili con quelli appena descritti, fatto salvo il fatto che:
a) i diodi di soppressione, previsti internamente per il controllo dei carichi induttivi, avranno ora gli anodi collegati tra loro, sempre convergenti verso il pin10 a essi dedicato, detto ora GND;
b) per funzionare sarà necessario fornire un'alimentazione esterna Vss sul pin9, la stessa richiesta dalla periferica da controllare.
La gestione software di questo progetto è piuttosto semplice: per poter ricostruire il carattere desiderato sulla matrice di LED bisogna in primo luogo predisporre una variabile contenente le 5 parole da trasferire sulle Righe, una per ciascuna delle 5 Colonne previste da questo visualizzatore; ciascuna parola avrà la dimensione di un byte e conterrà i bit necessari per tenere accesi (con 1) o spenti (con 0) i rispettivi LED della Colonna a essa relativa; poiché le Righe previste sono sette, il valore del bit più significativo (bit7) di ciascuna parola è irrilevante e, per precisa scelta, sarà tenuto a 0. La **figura 10** mostra (nei riquadri incorniciati di rosso) le cinque parole necessarie per visualizzare il carattere "0" e la **figura 11** mostra l'aspetto della variabile Tab_0 che le raccoglie, assieme alla prima parte di una nuova variabile, Tab_A, che raccoglie a sua volta le parole relative alle lettere dell'alfabeto.
Per mostrare il carattere "0" sul display sarà dunque necessario predisporre il primo byte della tabella Tab_0, 00111110B=3EH, sulla PortA e, subito dopo, scrivere la parola 00000001B=01H sulla PortB: immediatamente le linee forzate a 1 della prima porta accenderanno i LED collegati a catodo comune sulla prima Colonna (quella a sinistra del display), messa a massa dal bit0=1 invertito dal canale ULN2003 a esso collegato.



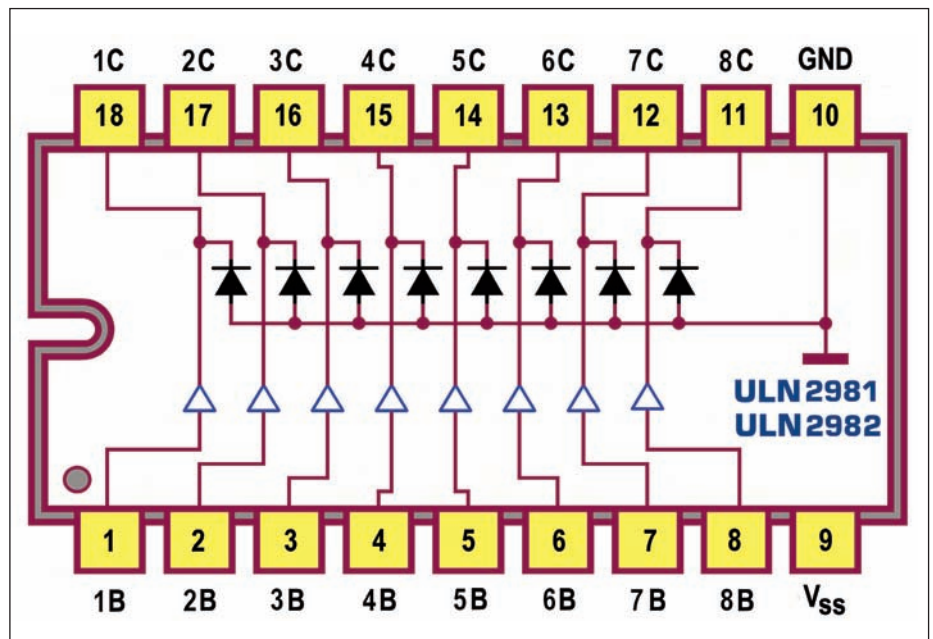
*Figura 8: High-voltage/high-current Darlington transistor arrays ULN2803: pinout.*



*Figura 9: High-voltage/high-current Darlington transistor arrays ULN2981 (non invertente): pinout.*

DataOUT0,A → R1 →
DataOUT1,A → R2 →
DataOUT2,A → R3 →
DataOUT3,A → R4 →
DataOUT4,A → R5 →
DataOUT5,A → R6 →
DataOUT6,A → R7 →
DataOUT7,A ........ →
C1 C2 C3 C4 C5
DataOUT0,B → → →
DataOUT1,B → → →
DataOUT2,B → → →
DataOUT3,B → → →
DataOUT4,B → → →
DataOUT5,B — n.c.
DataOUT6,B — n.c.
DataOUT7,B — n.c.

PortA (bit0) → 0 1 1 1 0 R1
PortA (bit1) → 1 0 0 0 1 R2
PortA (bit2) → 1 0 0 1 1 R3
PortA (bit3) → 1 0 1 0 1 R4
PortA (bit4) → 1 1 0 0 1 R5
PortA (bit5) → 1 0 0 0 1 R6
PortA (bit6) → 0 1 1 1 0 R7
PortA (bit7) → 0 0 0 0 0 .....
CCCCC
12345
PortB (bit0) → 1 0 0 0 0 0 0 0
PortB (bit1) → 0 1 0 0 0 0 0 0
PortB (bit2) → 0 0 1 0 0 0 0 0
PortB (bit3) → 0 0 0 1 0 0 0 0
PortB (bit4) → 0 0 0 0 1 0 0 0
PortB (bit5) → 0 0 0 0 0 0 0 0
PortB (bit6) → 0 0 0 0 0 0 0 0
PortB (bit7) → 0 0 0 0 0 0 0 0

*Figura 10: valori binari da predisporre, in sequenza, sulle due porte del microcontrollore.*

A questo punto il programma deve attendere un breve istante (dell'ordine del millisecondo) e poi provvedere alla stessa operazione sul secondo byte della tabella Tab_0, 01010001B=51H, scrivendolo sulla PortA e rendendone visibile il contenuto sulla seconda Colonna, forzata a massa dal codice 00000010B=02H scritto su PortB subito dopo; dopo aver consumato un altro millisecondo la medesima operazione sarà condotta sul terzo byte 01001001B=49H di Tab_0 (attivato forzando 00000100B=04H su PortB), e così via per il quarto byte (01000101B=45H su 00001000B=08H) e per il quinto (00111110B=3EH su 00010000B=10H).

In sostanza dopo poco più di 5 ms tutte le colonne sono state interessate dall'informazione necessaria per mostrare un "0" e durante questo rapido istante sarà già possibile intuirne la forma; ma solo ripetendo a oltranza tutti i comandi descritti (imponendo una commutazione continua tra le colonne, sufficientemente veloce da garantire l'effetto di persistenza ottica) il carattere sarà reso stabilmente visibile.

In conclusione possiamo sottolineare che la natura e l'aspetto modulare di questi display si presta al loro utilizzo in sequenze di più di uno, consentendo di pilotare più di una matrice di LED; per ciascuna di esse sarà necessario mettere in conto 5 Colonne (essendo comunque le Righe condivise tra di loro) e questo fatto può complicare il progetto e richiede necessariamente un numero maggiore di porte, non sempre disponibili.

Per organizzare visualizzatori con due o più matrici di LED è dunque necessario studiare modifiche circuitali in grado di rendere minimo il numero di linee d'uscita; ma di questo ci occuperemo in una prossima occasione. ❏

```
        ;-----------------------------------------
Tab_0   DB   00111110B   ;  "
        DB   01010001B   ;  "
        DB   01001001B   ;  "
        DB   01000101B   ;  "
        DB   00111110B   ;  "
        ;-----------------------------------------
Tab_A   DB   01111110B   ;  "
        DB   00001001B   ;  "
        DB   00001001B   ;  "
        DB   00001001B   ;  "
        DB   01111110B   ;  "

        DB   01111111B   ;  "
        DB   01001001B   ;  "
        DB   01001001B   ;  "
        DB   01001001B   ;  "
        DB   00110110B   ;  "

        DB   00111110B   ;  "
        DB   01000001B   ;  "
        DB   01000001B   ;  "
        DB   01000001B   ;  "
        DB   00100010B   ;  "
```

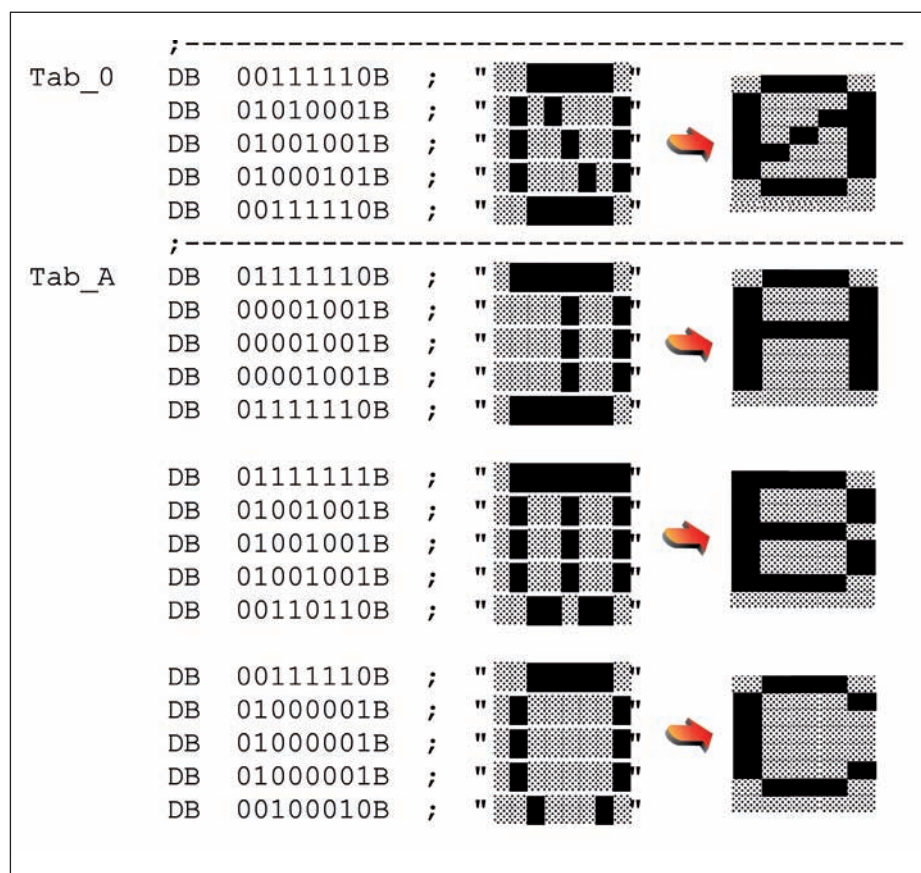



*Figura 11: variabile Tab_0 con i 5 bytes del carattere "0", e un estratto della Tab_A con tutte le lettere dell'alfabeto.*

CODICE MIP 2830213

di DANIELE CAPPA

# IL TRAFFICO di un radioamatore

**Alcune pagine che riguardano il traffico di un radioamatore, una breve chiacchierata sulle domande più comuni rivolte da un "non radioamatore".**

Siamo a metà febbraio di alcuni anni fa... la TV è i bianco e nero, la benzina è sotto le 300 lire al litro. Il luogo, seconda collina del torinese, da cui si vede la parete posteriore della Basilica di Superga, quella su cui il 4 maggio del 1949 si schiantò l'aereo del Grande Torino. Quella sera, finita la cena, è stato proposto a noi ragazzi di scendere in quello che era il magazzino / laboratorio a vedere una cosa. Era ovviamente un ricetrasmettitore dall'aspetto imponente, un Sommerkamp TS5024P (in CB ovviamente), con il mobile in legno, un orologio elettrico e tante lucine, troppe per noi ragazzi.
Siamo rimasti ad ascoltare le voci che uscivano da quella strana radio fino a che si è fatto tardi, ci siamo rivisti la sera dopo e abbiamo iniziato a vincere la paura e a premere anche noi il pulsante sul microfono, quasi stupiti che quella cosa trasmettesse per davvero. Per fortuna abbiamo trovato una persona che ha perso alcune ore del suo tempo a spiegarci che cosa stavamo facendo, dove eravamo e come funzionavano le cose in questo mondo per noi nuovo. Non so chi sia, non ho mai saputo il suo nome e non l'ho mai più sentito. Certamente il merito, o la colpa, se a distanza tanti anni sono ancora qui a "giocare con le radio", è anche sua.
Lo scopo di queste righe è rispondere a due domande banali che immancabilmente vengono rivolte ai radioamatori: "Che cosa vi dite?" e "Quanto tira la tua radio?" (un collega radioamatore ha usato questa frase come titolo del suo sito. Bellissimo! Grazie Edoardo). Ma non è cosa facile, così come non è facile spiegare al non fumatore perché si fuma, al non viaggiatore perché si viaggia o al non appassionato di fai da te perché ci si ostina a fare in casa qualcosa che si trova già fatto!
Il radioamatore è definito "OM", old man, uomo vecchio. Definizione internazionalmente accettata di "radioamatore patentato", spesso male utilizzata per distinguerlo dal CB (Citizen Band) il quale, "giocando" anch'egli con la radio, potrebbe essere lecitamente chiamato anch'egli radioamatore (ma non entriamo in simili disquisizioni).

## DUNQUE: "CHE COSA VI DITE?"

Tu al telefono di che cosa parli? Quando vai in pizzeria con gli amici di che cosa parli? Ovviamente di qualsiasi cosa. Così io in radio!
Ci si trova alla sera con gli amici, sempre nello stesso posto, dal 19xx, si parla di quel che capita. Esistono regole, e determinati argomenti sono tabù, per legge o per semplice educazione. Nulla di diverso da Skype, insomma! O da qualsiasi altro "luogo virtuale". Non proprio, io ho solamente bisogno della radio, le pile cariche o l'auto. Non dipendo da nulla di esterno, o meglio da nulla che non sia sotto il mio diretto controllo. Può essere un'affermazione determinante, anche se (per fortuna) di solito è assolutamente irrilevante.
Questo è il "traffico serale" con gli amici, prima o dopo cena, qualcuno è in casa, altri in auto. Situazione analoga si è già presentata al mattino, quando chi va al lavoro è in auto, chi ha orari diversi ed è a casa, con la radio accesa prima di uscire. La platea è eterogenea, spesso in queste situazioni ci si conosce da anni e spesso ci si conosce di persona.
Si ma... a che distanza? Non moltissima, di solito entro i 100 Km, più probabilmente il corrispondente è nell'ambito cittadino, e comunque entro i 20-30 km. Qui si impiegano apparecchietti semplici ed economici, il traffico si svolge in VHF o, al più, in UHF, ovvero in gamma due metri (145 Mhz) oppure 70 cm (435 Mhz). Prevalentemente in FM con RTX veicolari (magari utilizzati in casa), oppure con RTX palmari (spesso cinesi da poche decine di euro). Piccole antenne verticali, da 50 cm a 5 metri circa.
In questa situazione si passa la maggior parte del tempo, ma lo scopo non è certamente questo, nella chiaccherata serale si parla del nuovo progetto, dell'antenna da mettere in cantiere, o dell'altro traffico. La maggior parte delle volte lo scopo non è dire qualcosa, ma poterlo dire.
Chi si dedica al "collegamento", ovvero a realizzare un collegamento bilaterale interessante, vuoi per la distanza o per la difficoltà a realizzarlo, non ha alcun interesse a dire qualcosa, dunque le frasi sono ritualizzate con uno schema simile a questo:

• io chiamo;
• tu rispondi;
• ti dico il nome, dove sono, come ti sento, nel caso qualche parola sulla stazione;
• mi ricambi la cortesia e mi passi nome, ecc.;
• ti saluto;
• mi saluti.

Fatto! Il tutto è durato da qualche decina di secondi a qualche minuto. Del resto il corrispondente potrebbe non conoscere la nostra lingua, il mio inglese è poverino e probabilmente lo è anche il suo, per questo di solito si ricorre a frasi prevedibili e note a tutti.

## TUTTO QUI? E QUAL È LO SCOPO?

In pratica sì, è tutto qui. lo scopo è realizzare il collegamento, non è sempre facile e spesso per ottenere lo scopo si ricorre a soluzioni molto diverse, ovvero non è "che cosa ci diciamo" ma "come lo diciamo".
Una buona antenna è determinante, meglio se posta in un luogo favorevole. Ma non tutti abitano su una collina, o in cima a una vetta innevata! Una buona antenna messa in giardino serve a poco.
Come sempre è una questione di dB. Una buona antenna guadagna dB, una buona posizione fornisce attenuazioni minori. Un trasmettitore con una potenza elevata aiuta.
Le gamme adibite al servizio di radioamatore partono della soglia delle onde medie (1850 kHz, 160 metri, con una fettina anche a 136 Khz, in piena gamma onde lunghe dove è consentito un solo watt ERP), fino alle microonde, 250 GHz, con 500 W su tutte le gamme intermedie. La potenza di un RTX recente si attesta di solito sui 50 W per le VHF e UHF, 100 W per le gamme più basse (HF), le onde corte quindi da 1,8 a 29 MHz.
È facile immaginare che frequenze così diverse abbiano comportamenti assolutamente differenti. Dunque il collegamento banale in 20 metri è una meta da ricordare se effettuato in microonde. Il record mondiale in gamma 13 cm, 2.400 Mhz (vicina alla gamma Wi-Fi, tanto per capirci, è del gennaio 2011:
*Congratulations to both Adrian VK4OX and ZL1TPH Stephen for their QSO on 13 cm SSB this afternoon, spanning 2317.1 km!*
Il traffico a lunga distanza (il DX) è governato dall'alto. Senza ricorrere a interventi divini l'aiuto che rende possibile il collegamento bilaterale oltre la linea dell'orizzonte è la parte alta della ionosfera.
Questa è divisa in strati, ogni strato ha altezza e caratteristiche diverse che lo rendono trasparente a determinate frequenze, mentre per altre si comporta come uno specchio riflettendo nuovamente verso terra il segnale che lo raggiunge.
Le condizioni degli strati dipendono dalle condizioni del sole, dalla presenza o meno di questo (quindi dal giorno o dalla notte), dall'ora, dalla stagione. Insomma con gli anni si può avere una idea se una determinata banda è aperta o meno (ovvero se è utilizzabile per il collegamento distante). Ma la certezza la si ha solo accendendo la radio.
In onde corte la gamma dei 20 metri (14 MHz) fornisce le migliori possibilità, con una certa uniformità durante l'anno (ma in inverno si chiudono anche queste). Scendendo verso il basso aumenta il rumore e crescono le dimensioni delle antenne, salendo verso l'alto (come frequenza, ovviamente) diminuiscono le dimensioni delle antenne, e anche le probabilità di trovare la "banda aperta". Dunque un collegamento relativamente comune in 20 metri lo è meno in 10 metri.
Salendo ancora le cose si complicano e le difficoltà salgono in modo esponenziale. Un collegamento tra i due estremi del nostro paese non è neppure preso in considerazione in 20 metri, ma diventa appetibile in gamma 2 metri.
A questo si aggiungono sistemi specializzati, che fanno a meno della ionosfera, dunque traffico via satellite o meteor-scatter, sfruttando la scia ionizzata che si lascia dietro una meteora, una stella cadente.

## ALLORA CHE SI FA? RADIO & PC!

Si fa come si può, ognuno di noi sceglie la strada che più gli è congeniale e decide se utilizzare potenze elevate, frequenze al limite basso delle HF, microonde, il microfono, il tasto o il computer.
Da molti anni si effettua traffico digitale, ovvero codificato a mezzo di macchinette, in ambito militare (negli USA ovviamente) si è diffusa la telescrivente (da ora in avanti RTTY), che utilizza un codice a 5 bit, il Baudot. I primo link (militare) risale al 1932. Nel primo dopoguerra alcune di queste macchine si sono rese disponibili nel mercato dell'usato, come surplus, recupero, roba da buttare insomma.

Figura 1: macchina telescrivente meccanica, il modello 15.

In RTTY il collegamento tra due stazioni avviene in modo seriale asincrono, i cinque bit che compongono il codice Baudot si presentano alla macchina ricevente uno alla volta, in sequenza, e sono preceduti da un bit di start e seguiti da uno di stop (con alcune varianti dovute al tipo di macchina usata), affinché la macchina possa capire che la sequenza che segue rappresenta un carattere da stampare. Questa è una procedura "asincrona", ogni singolo carattere ha in sé gli elementi che lo dividono dai suoi simili, non esiste quindi alcun vincolo di spaziatura tra due caratteri successivi; così come non esiste alcun metodo per recuperare un carattere perso. Correzione di errore, Viterbi e simili erano là da venire. Del resto le macchine erano più simili a una macchina per scrivere elettrica, assomigliavano di più a una macchina per cucire che a un computer. Dal peso considerevole, erano montate su un tavolo ammortizzato e su un tappeto di gomma spesso alcuni centimetri per impedire che "camminassero da sole".

Per noi abituati a velocità che si misurano in Mbit quelle velocità appaiono strane, ridicole. Questa telescrivente funzionava a 50 baud, il paragone con i vecchi modem telefonici a 56 kb è facile, non esattissimo, ma come ordine di idee ci siamo. Trasmettevano, e i software moderni lo fanno ancora, due toni audio, mark e space, ovvero ognuno dei due stati logici era trasmesso con un tono audio diverso. Il ricevitore era dotato di filtri che separavano i due toni ricostruendo il segnale binario originale. L'interfaccia verso la macchina meccanica era un comando elettrico attuato con una corrente costante a 20 mA opportunamente interrotta. Questa situazione è rimasta sostanzialmente invariata per una ventina di anni, più o meno.

Poi con gli anni la cosa si è evoluta, con i primi computer è stato possibile eliminare la macchina meccanica, pur conservando il demodulatore (ora finalmente a filtri attivi). Di che anno parliamo? Metà anni Ottante, più o meno. Software per telescriventi era disponibile anche per i piccolini dell'epoca, VIC20 e C64, ma anche per Spectrum e per i primi compatibili IBM, ancora sotto DOS evidentemente, e con prestazioni che le vecchie meccaniche neppure immaginavano.

Qualche anno e buttiamo anche il demodulatore. Con il solo aiuto di uno squadratore esterno il software poteva (quasi) sostituire il demodulatore ed effettuare una decodifica decente.

## POI LA SVOLTA: LA SCHEDA AUDIO

Già, buona parte del traffico amatoriale non in fonia si svolge oggi grazie al DSP della scheda audio del PC. A volte là dentro si trova una "quasi radio".

Ma che centra ora il PC con la radio? Fino a qualche anno fa per effettuare un collegamento, a parte la radio ovviamente, potevi aver bisogno del microfono, del tasto telegrafico o della telescrivente. Volutamente escludiamo un attimo la TV a scansione lenta (SSTV) giusto per non complicarci la vita.

Poi il PC ha introdotto un altro fattore, la citata scheda audio, in grado di effettuare campionamenti e trattare quindi i segnali audio come informazioni digitali.

La svolta è relativamete recente, alla fine del 1998 un radioamatore inglese, Peter Martinez G3PLX, basandosi sul lavoro di Pawel Jalocha (SP9VRC con SLOWBPSK) mette a punto un "modo digitale" che fa letteralmente impallidire i vecchi sistemi, ovvero la vecchia telescrivente con il suo codice a 5 bit.

Le mie prime prove sono di qualche mese dopo. Io e Leo, e il risultato era folle: quel software decodificava perfettamente il segnale audio anche se ci si parlava sopra! Una cosa impensabile, in RTTY basta un soffio per perdere alcuni caratteri e qui potevi parlare mentre il microfonino del PC stava ricevendo, senza neppure una connessione diretta alla radio!

La velocità è ancora più bassa della vecchia telescrivente, poco più di 31 baud, ma per contro la larghezza di banda è davvero minima e su un solo canale audio, che in SSB è già molto contenuto (2.400 Hz circa), ci stanno contemporaneamente decine di comunicazioni indipendenti, ovvero decine di coppie di corrispondenti che svolgono la loro attvità senza darsi fastidio a vicenda. Al contrario della vecchia RTTY, che utilizzava una modulazione FSK, qui è utilizzata una modulazione di fase (PSK) con una differenza di 180 gradi tra i due stati.

Il nuovo modo si afferma in fretta, e malgrado la comparsa negli anni di altri modi di ancora più performanti, il PSK31 resiste agli assalti della concorrenza senza battere ciglio.

il problema è infatti il rapporto tra segnale e rumore (SNR), il PSK31 arriva a –11dB, altri modi più recenti scendono ancora più in basso.

Per l'ultimo (a mia conoscenza) abbiamo anche la data di nascita. Il modo è ROS, l'autore è il Dottor José Alberto Nieto Ros, ingegnere in telecomunicazioni presso l'università "Alfonso X Sabio", dunque non è certamente l'ultimo arrivato. La ricorrenza è il 18 febbraio 2010. Qui si scende a livelli impensabili fino a pochissimi anni fa, siamo sotto i –20 dB. Significa che un collegamento possibile in PSK31 utilizzando 100 W diventa possibile (nelle stesse condizioni) in ROS4 utilizzando solamente 1 W. Se aggiungiamo che il PSK era già a –11 dB...

Qui potremmo elencarne moltissimi, alcuni sono comparsi e spariti nel giro di pochissimo tempo e, pur offrendo prestazioni notevoli, non hanno avuto la diffusione che meritavano. Il PSK31 resiste e oggi è affiancato dal JT65. Accompagnato da un software più fruibile (JT65HF) e meno complicato da usare, sta dando molte soddisfazioni.

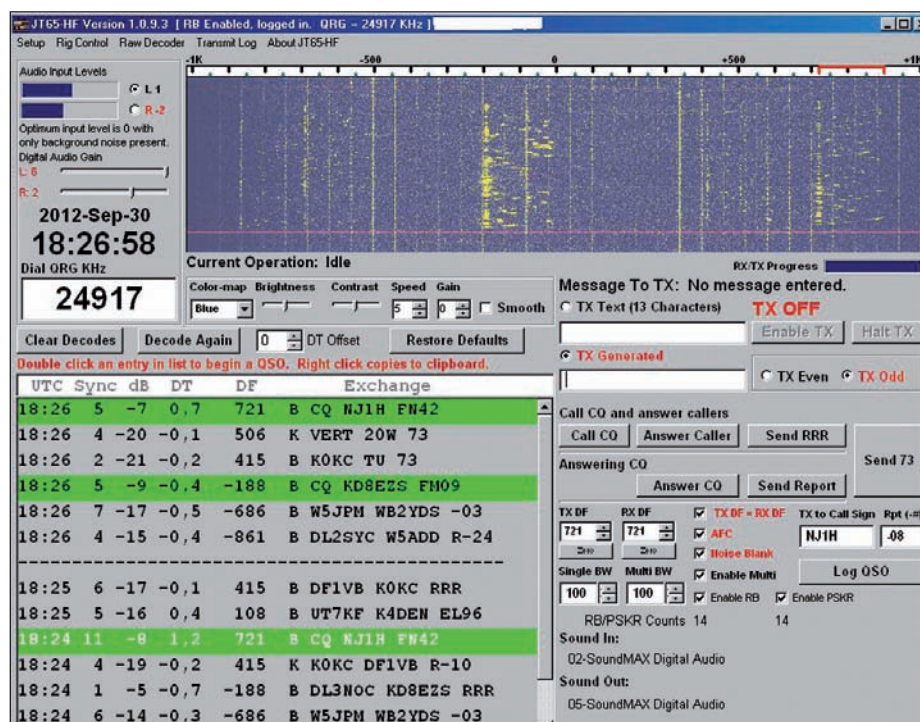



*Figura 2: schermata del software JT65HF.*

Figura 3: cartoline "radiofoniche".

Questo protocollo ha molti fratelli, si avvale di una trasmissione estremamente lenta, in 50 secondi di trasmissione continua sono trasmessi solo pochissimi caratteri (al massimo 13), prevedendo un protocollo di correzione di errore raggiunge i –24 dB, ovvero decodifica quando l'orecchio non sente altro che rumore. Il sistema è stato pensato per traffico via luna, ovvero utilizzando il nostro satellite come un enorme (e lontanissimo) specchio. L'utilizzo di questo software in onde corte risale al 2006 grazie a due radioamatori, un russo e un giapponese.
Ci siamo distratti divagando su sistemi di utilizzo della radio diversi dal solito, dunque non la chiacchierata serale a cui molti pensano. Sistemi che danno una mano a raggiungere l'obbiettivo, che rimane la realizzazione del collegamento. Ecco perché non si utilizza il cellulare.
Il raggiungimento dello scopo è soggetto alle opinioni e ai gusti personali, dunque alcuni preferiscono il modo meno performante, anche se è richiesta più potenza, altri preferiscono pochissima potenza utilizzando le potenzialità fornite dal software utilizzato. Altri ancora preferiscono le orecchie, utilizzando la telegrafia e potenze a volte irrisorie, spesso al di sotto del watt. È evidente che in situazioni diverse anche i risultati saranno molto diversi, ma la soddisfazione personale per aver effettuato un collegamento non eccezionale, ma utilizzando un apparecchietto minuscolo, probabilmente autocostruito, è impareggiabile. Basterà dare in pasto a Google "RTX QRP" oppure "RTX QRPP" e se ne trovano a migliaia.
La quasi totalità del traffico in modi digitali su svolge con potenza basse, mediamente 20 W. Alcuni si dedicano al QRP, ovvero al traffico con basse potenze, 5 W, ma altri più fondamentalisti scendono ancora e si attestano su qualche centinaio di milliwatt.

## LA CONFERMA

Accompagnano questo articolo alcune immagini che inequivocabilmente sono cartoline. Di che si tratta? Alla fine dell'avvenuto collegamento, la naturale conclusione è la QSL, la "cortesia finale" ovvero la conferma cartacea che il collegamento è realmente avvenuto. La cartolina contiene il nominativo dei due corrispondenti, spesso l'indirizzo (dati che sono stati in parte eliminati), la data, il modo e la banda in cui è stato effettuato il collegamento.
Nello specifico queste sono in realtà delle e-qsl, ovvero delle cartoline elettroniche, diffuse da un sito che si occupa di smistarle e recapitarle agli interessati. Questo tipo di conferma non ha valore legale, ma a me poco importa. Lo standard prevede l'invio di una vera cartolina cartacea spedita via "bureau". Le tante associazioni di radioamatori si incaricano di ricevere, smistare e inoltrare alle altre associazioni le cartoline che in questo modo raggiungono qualsiasi parte del mondo (o quasi...) a costo zero, anche se i tempi non sono rapidissimi. L'alternativa è "direct" ovvero diretta, via posta tradizionale, con francobollo adatto e data in pasto al locale servizio postale.
Da questo punto di vista la cartolina virtuale non ha rivali, è immediata, non occupa spazio, non costa nulla ma serve solo alla soddisfazione personale. Non si può avere tutto!

## PROVIAMOCI ANCHE NOI!

Abbiamo effettuato un rapido volo radente su alcuni modi digitali, vediamo ora come questi si presentano sul monitor del PC, come si presentano al nostro orecchio, ovvero "che cosa si sente". Nella **figura 4** si osserva la schermata in PSK31 di "Mixw2" software multimodo in grado di ricevere e trasmettere in molti modi diversi. Ve ne sono altri, alcuni gratuiti, altri no, sia per Windows sia per Linux. Non manca neppure il necessario per Android! Partiamo dal Digipan, poi Multipsk, MMTTY, il multipiattaforma FLdigi e via di seguito.

Figura 4: schermata del software Mixw2 PSK31.

militari verso la fine dell'ultimo conflitto, all'inizio per evitare i disturbi nei confronti dei primi sistemi radar, poi per passare alle comunicazioni vere e proprie. Il segnale occupa una banda molto più ampia di quella necessaria a trasmettere l'informazione. A paragone con il PSK31, ROS esso ha metà velocità e una larghezza di banda decine di volte maggiore. A prima vista sembra uno spreco ma è il prezzo da pagare per avere vantaggi da altri punti di vista. Abbiamo una buona immunità ai disturbi a banda stretta, la ricostruzione del segnale "sparpagliato" durante la codifica "sparpaglia" il segnale disturbatore riducendone drasticamente gli effetti negativi.
Alcuni software si appoggiano a servizi sul web, JT65 ha più di un web report, ovvero una pagina web a cui il software comunica chi e come riceve. Alcune pagine hanno una buona interfaccia grafica, altre forniscono solamente un elenco di stazioni con il segnale ricevuto e pochi altri dati.
Se avete piacere di sentire dal vivo quanto hai letto, allora raggiungete questo indirizzo:

http://websdr.ewi.utwente.nl:8901/

Troverete un "web SDR" ovvero una radio basata su un computer. Si trova in Olanda, con lo zampino di un istituto universitario.
SDR è l'acronimo di "Software Defined Radio", ovvero un ibrido in cui la radio quasi non c'è, si riduce a un convertitore che porta la frequenza che si desidera ricevere a valori molto bassi, una sorta di media frequenza audio. Questo segnale (con due fasi diverse) viene dato in pasto alla scheda audio del PC. Software idonei si occupano di "trattare" il segnale e di estrarne l'informazione. È evidente dunque che il modo di ricezione, la banda passante e qualsiasi altra cosa ci venga in mente è realizzabile adeguando il software del computer. Dietro alle "web SDR", come ad altre "web radio" c'è un sistema analogo. Spesso (il link indicato di sicuro) è richiesta la presenza di java sul PC dell'utente.
Un ricevitore SDR è realizzabile

La quasi totalità del software dedicato ha una schermata di sintonia denominata waterfall (cascata) perché scorre dall'alto verso il basso rappresentando da sinistra verso destra le frequenze audio ricevute dalla scheda audio del PC, mentre lo spostamento verso il basso visualizza solo le ultime decine di secondi ricevute.
Ogni modo ha le sue particolarità, alcuni molto validi sono stati abbandonati, o sono poco usati rispetto alle potenzialità. Vuoi perché il software necessario non è così immediato nell'uso, vuoi perché per effettuare un collegamento è necessario essere in due, e se chiamo in un modo in cui nessuno risponde, prima o poi si cambia sistema a favore di uno magari meno performante ma più affollato.
In questo modo sono caduti nel dimenticatoio sistemi validi: lo stesso ROS e altri del gruppo che utilizzano lo "Spread Spectrum" stanno subendo un calo di popolarità. Due parole sullo "Spread Spectrum": si tratta di un sistema nato per applicazioni

Waterfall-JT65A

Waterfall-olivia-500-16

Waterfall-ROS

Waterfall-RTTY-170

Figura 5: schermate di sintonia chiamate "waterfall".

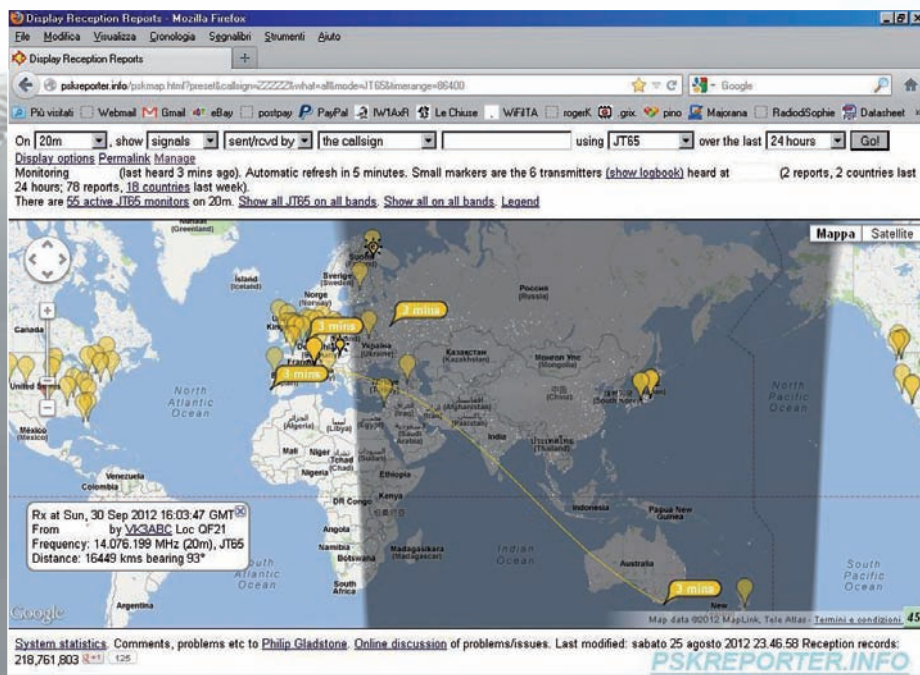



*Figura 6: WEB report JT65.*

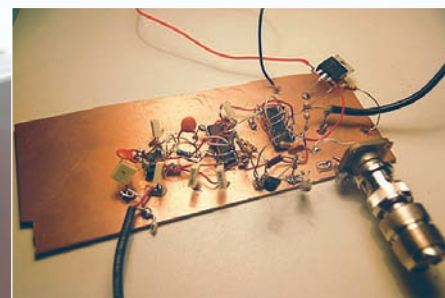



*Figura 7: un semplice ricevitore SDR.*

con pochissima spesa, funziona in modo eccellente, ma richiede una buona scheda audio. Essendo questa l'elemento che supporta il tutto, un device audio di bassa qualità fornirà risultati pessimi, mentre una scheda di alto livello fornirà ottime performance anche se l'hardware esterno è ridotto all'osso.

È dunque possibile effettuare ascolti in modalità digitale utilizzando uno dei software citati e un web SDR, utilizzando il nostro solito PC.

## SPENDIAMO DUE MINUTI E VEDIAMO COME FARE:

Scarichiamo MIxW (Google, aiutaci tu), qualsiasi versione, fino alla 2,18 l'installazione è facilissima.

Installiamo il programma, anche se funzionerà in modalità demo e ci farà aspettare alcuni secondi tra un cambio di modo e l'altro. Lanciamo il programma, ci chiederà di inserire un nominativo, qualsasi cosa va bene, A1AAA è perfetto, tanto non possiamo trasmettere.

Quindi, "configure" e "Input volume" spuntiamo nel mixer di Windows il "missaggio stereo", oppure l'ingresso "wave" e qui le cose potrebbero cambiare da una scheda audio e l'altra. Al peggio utilizzeremo il microfono sfruttando l'audio che proviene dalle casse del PC che verrà ridigerito dal microfonino del medesimo. Non è il massimo ma l'immunità del sistema al rumore è efficace fino a permetterci anche simili nefandezze.

Andiamo all'indirizzo del web SDR e ci posizioniamo a 14,070, modo USB. Dovremmo sentire una serie di "toni traballanti".

Torniamo sulla schermata di MixW, poniamo il "mode" su PSK31, posizioniamo il puntatore del mouse su una di quelle righe verticali e ci clicchiamo sopra.

Fatto, ora il programma decodifica. Il modo è PSK31 (il modo selezionato appare in basso a destra) e la decodifica avviene nella parte centrale dello schermo, come nella **figura 4** riportata.

Spostandoci di poco verso l'altro troveremo segnali RTTY, basterà cambiare nuovamente il modo al programma e posizionare il cursore tra i due toni. In questo le figure 2 e 4 potranno esserci di aiuto.

A 14.076 possiamo provare JT65HF, la configurazione è assolutamente analoga a quella descritta, l'unica cosa da fare è attendere qualche minuto che lo schermo ci mostri quanto ricevuto. La decodifica qui avviene dopo il cinquantesimo secondo del minuto in corso, dunque è indispensabile che l'orologio del PC sia perfettamente sincronizzato!

In entrambi i casi il volume dell'ingresso andrà regolato fino a ottenere una schermata scura con i tratti più chiari che non dovranno sconfinare nel giallo o, peggio, diventare rossi.

Siamo alla fine della nostra chiacchierata, l'argomento è vasto e le possibilità sono molte, e crescono in continuazione. Chi pensa che quella del radioamatore sia un'attività morta, o moribonda è male informato.

È evidente che l'impegno è grande, e che comunque il tutto deve trovarsi nel nostro campo di interesse. Non era mia intenzione convincere nessuno, sembrava giusto spiegare a chi ha avuto la pazienza di seguirmi in che cosa consiste il traffico svolto oggi da un OM. Lasciandoci indietro reminiscenze CB (che abbiamo tutti, ma cambiando il luogo deve cambiare anche il comportamento), comprese le esclamazioni e definizioni folcloristiche tanto in voga in 27, quanto malviste altrove. ❑

**CODICE MIP 2830217**

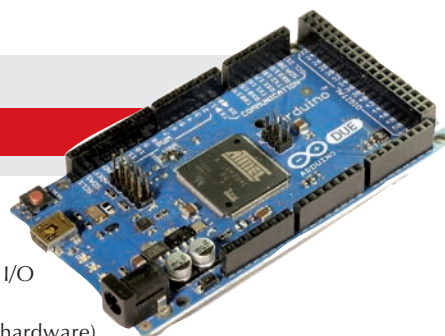
IEshop

# Lo shop dei prodotti Inware Edizioni...

## Nuova piattaforma!

PRIMO PIANO

### Arduino Due

Arduino Due, la nuova arrivata nella famiglia Arduino, è la prima scheda basata su un processore a 32 bit ARM, l'Atmel SAM3X8E ARM Cortex-M3, che migliora tutte le funzionalità standard di Arduino e aggiunge nuove caratteristiche. Possiede 54 pin di I/O digitali (16 dei quali possono essere utilizzati come uscite PWM, con risoluzione selezionabile), 12 pin di input analogici a 12 bit di risoluzione, 4 UART (porte seriali hardware), e due uscite DAC (digital to analog converter), un cristallo oscillatore a 84 MHz , due connessioni USB, un jack di alimentazione, un header ICSP, un header JTAG e un pulsante di reset. La tensione massima che i pin I/O sono in grado di fornire o tollerare è di 3,3 V (tensioni più elevate, come 5 V al pin di input potrebbero danneggiare la scheda). Arduino Due dispone di due connettori USB, il connettore micro-usb B è quello nativo in grado di agire come host USB, ovvero è possibile collegare periferiche esterne compatibili con USB, come mouse, tastiere, smartphone; mentre l'altra porta USB con il connettore di tipo A è per scopi di debugging.

***CODICE: A000062 PREZZO: € 47,19***

PRIMO PIANO

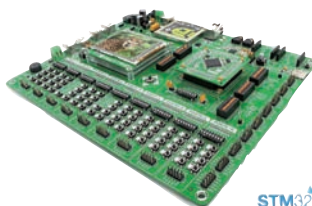
### Arduino Starter Kit

Questo nuovo kit vi introdurrà nel mondo di Arduino in maniera molto pratica ed è perfetto per imparare e sperimentare le conoscenze dell'elettronica. Imparerete il facile utilizzo di Arduino mediante la costruzione di diversi progetti creativi.
Il kit comprende una selezione dei componenti più comuni e utili e un libro con le istruzioni per 15 progetti pratici. Iniziando dalle basi dell'elettronica, fino a progetti più complessi, questo kit vi aiuterà a controllare il mondo fisico con sensore e attuatori.

***CODICE: K000007 PREZZO: € 96,68***

### Scheda di sviluppo per STM32 ARM

EasyMx PRO v7 for STM32 ARM è una piattaforma di sviluppo completa per i microcontrollori ARM Cortex-M3 and Cortex-M4. Contiene molti moduli on-board necessari per lo sviluppo di una vasta varietà di applicazioni, incluso il multimediale, Ethernet, USB, CAN e altro. Il programmatore e il debugger on-board supportano oltre 180 microcontrollori ARM. La piattaforma è venduta con la MCU STM32F107VCT6. EasyMx PRO v7 for STM32 ARM ruota intorno alla connettività. Avendo due connettori differenti per ogni porta, sarete in grado di connettere schede accessorie, sensori o le vostre schede custom in modo più facile che mai. Il potente programmatore on-board mikroProg e il debugger hardware possono programmare e fare il debug di oltre 180 dispositivi STM32. Ne avrete bisogno, sia che voi siate neofiti o professionisti.

***CODICE: EASYMX7-STM32 PREZZO: € 156,09***

NOVITÀ

NOVITÀ

### Mikromedia per STM32

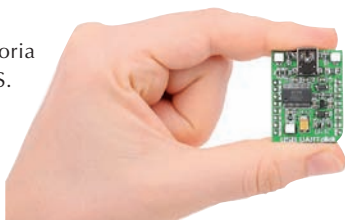
Questa scheda fornisce una piattaforma di sviluppo multimediale d'alta qualità completa per il dispositivo STM32F407VGT6. Possiede numerosi moduli on-board che permettono di scrivere applicazioni multimediali. Questa scheda può essere usata sia per lo sviluppo o come prodotto finale. Il microcontrollore STM32 on-board è pre-programmato con un veloce bootloader USB HID e, quindi, pronto a lavorare in brevissimo tempo. Non dovete spendere un euro in più sui programmatori. Per coloro i quali ne necessitano, la scheda può, inoltre, essere programmata utilizzando un programmatore e debugger esterno mikroProg for SMT. Mikromedia for STM32 M4 è pienamente supportato dai compilatori per ARM mikroC, mikroBasic e mikroPascal. I compilatori sono forniti di dozzine di esempi che dimostrano l'utilizzo di ogni caratteristica della scheda stessa.

***CODICE: MKR-STM32-M2 PREZZO: € 91.96***

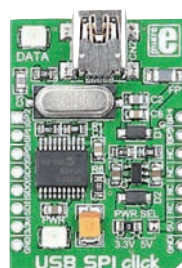
## USB UART click

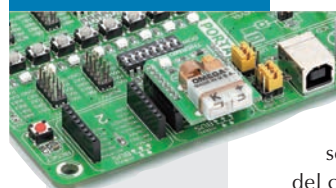
NOVITÀ

USB UART Click è una scheda accessoria che utilizza il nuovo socket mikroBUS. È una soluzione semplice e compatta per aggiungere comunicazione UART seriale via cavo USB. È provvista di modulo interfaccia FT232RL USB-to-UART e connettore USB MINI B. USB UART Click comunica con la scheda target via interfaccia UART. La scheda è alimentata a 5 V ma è progettata per usare sia livelli di I/O a 3,3 V sia a 5 V (selezionabili via jumper SMD). Possiede un diodo LED (verde) che indica la presenza di alimentazione. La scheda contiene i diodi LED RX e TX che indicano la trasmissione e la ricezione dei dati via USB. La scheda è usata in comunicazione RS232 con modem, stampanti, applicazioni PC e vari dispositivi che usano il protocollo UART.

***CODICE: CLICK-UART PREZZO: € 14,52***

## USB SPI click

USB SPI Click è una scheda accessoria che utilizza il nuovo socket mikroBUS. È una soluzione facile e compatta per aggiungere comunicazione seriale SPI via cavo USB. Essa è provvista di protocollo convertitore USB-to-SPI MCP2210 con GPIO e connettore USB MINI B. USB SPI Click comunica con la scheda target via interfaccia SPI e in qualità di dispositivo master. È disponibile un jumper SMD per la selezione del voltaggio di alimentazione, 3,3 V o 5 V. Possiede un diodo LED (verde) che indica la presenza di alimentazione. La scheda può essere usata per comunicare con i dispositivi client SPI direttamente dal vostro PC, senza il bisogno di avere un microcontrollore.

***CODICE: CLICK-SPI PREZZO: € 48,28***

NOVITÀ

NOVITÀ

## THERMO click

THERMO Click è una scheda accessoria che utilizza il nuovo socket mikroBUS. È dotata del convertitore termocoppia-digitale MAX31855K oltre al connettore PCC-SMP per sonde termocoppia K-type. MAX31855K ha un range di temperatura incluso tra –270 e 1.372 °C con sensibilità di circa 41 μV/°C. Possiede un convertitore ADC a 14 bit built-in. THERMO Click comunica con la scheda target via interfaccia SPI (sola lettura). La scheda è progettata per usare solamente un'alimentazione a 3,3 V. È fornita di diodo LED verde che indica la presenza di alimentazione. THERMO Click con sonda termocoppia K-Type è l'ideale per applicazioni di monitoraggio, process-control, termostatiche e molto altro.

***CODICE: CLICK-THERMO PREZZO: € 27,83***

## Espansione per POKeys

Questa scheda permette l'espansione del numero di output di Pokeys usando PoExtBus. Incorpora 8 high current/voltage open collector output adatti per applicazioni quali lampade, relè ecc.

OFFERTA

**Caratteristiche**

Utilizzo: fino a 10 x schede PoExtBus-compatibili connesse in cascata usando il Pokeys External Bus.
Output ad alta corrente 8 x 40 V 400 mA.
Output open-collector con diodi di protezione per sovratensioni.
5 board wire to board ribbon cable inclusi per connessione con Pokeys.

***CODICE: POEXTBUSOC PREZZO: € ~~18,00~~ € 15,73***

## PoKeys56UT (versione con terminali a vite)

PoKeys56UT è un dispositivo USB di facile utilizzo che combina parecchi input e output e non richiede una complessa conoscenza di programmazione. Pokeys56 fornisce 55 I/O digitali (tolleranti 5 V), 7 input analogici (12 bit). Tutti gli I/0 sono controllati dal software fornito che permette sia di usare l'interfaccia utente grafica che l'interfaccia console avanzata. Le impostazioni possono essere salvate nel dispositivo in modo che nessun software speciale sia richiesto sul sistema target. Pokeys56 può essere usato come scheda di acquisizione dati industriale o CNC.

***CODICE: POKEYS56UT PREZZO: € ~~60,50~~ € 47,19***

OFFERTA

## PoKeys56E (versione senza morsettiere a vite)

PoKeys56E è un dispositivo Ethernet di facile uso che combina diversi input e output e non richiede una complessa conoscenza della programmazione. Fornisce 55 I/O digitali (5 V tolleranti), 7 input analogici (12 bit). Tutto l'I/O è controllato dal software fornito che permette sia di usare l'interfaccia grafica utente o l'interfaccia console avanzata. Le impostazioni possono essere salvate nel dispositivo, in modo tale che nessun software specifico sia richiesto sul sistema target. PoKeys56 può essere usato come board industriale, controllo CNC o acquisizione dati. Questa versione non è accompagnata dalle terminazioni a vite.

***CODICE: POKEYS56E PREZZO: € ~~62,92~~ € 54,45***

OFFERTA

IEshop

# Lo shop dei prodotti Inware Edizioni... ma non solo!

**OFFERTA**

## Arduino Mini Light

Arduino Mini Light è la versione Micro miniaturizzata della scheda Arduino (senza parte USB) sprovvista di connettore downside. Le dimensioni estremamente ridotte (30x18 mm) permettono una diminuzione dello spazio di ingombro non indifferente. Fa uso di ATMega168 con 16K di spazio programma ed è utilizzabile con convertitore USB/Seriale per la programmazione e l'aggiunta della porta USB.

**CODICE: A000004 PREZZO: € ~~22,09~~ € 15,73**

**BEST SELLER**

## Modulo display da 3.2"

uLCD-32PT(GFX) è un modulo LCD "SMART" da 3.2" all-in-one compatto ed economico che fa uso della più recente tecnologia LCD a matrice attiva (TFT) e dispone di un controller grafico PICASO-GFX2 integrato che fornisce funzionalità "stand-alone" a qualsiasi progetto. Il PICASO-GFX2 appartiene a una famiglia di processori che dispone di un motore virtuale soft core altamente ottimizzato: E.V.E. (Extensible Virtual Engine). EVE è processore virtuale proprietario ad alte prestazioni con un set di istruzioni byte-code estensivo ottimizzato per eseguire programmi compilati in 4DGL. 4DGL (4D Graphics Language) fu specificatamente sviluppato da zero per l'engine core EVE. È un linguaggio ad alto livello di facile apprendimento e semplice da capire, eppure abbastanza potente per affrontare tantissime applicasioni grafiche embedded. 4DGL permette allo sviluppatore di scrivere applicazioni ad un alto livello di sintassi simile per gli altri linguaggi più popolari quali BASIC, C e Pascal, eseguendole direttamente nel processore PICASO-GFX2 integrato nel modulo uLCD-32PT. Permette all'utente di assumere il pieno controllo di tutte le risorse disponibili sulla piattaforma hardware quali porte seriali, display LCD grafico, memory card uSD, pins I/O ecc. Questo elimina il bisogno di avere un processore/controller esterno per pilotare il modulo uLCD-32PT via comandi seriali. Fornisce all'utente il completo controllo sul modulo hardware permettendogli di sviluppare potenti applicazioni.

**CODICE: ULCD-32-PT PREZZO: € 79,86**

## Annate Firmware 2006/2011 + Fare Elettronica 2003-2011

Tutti i numeri di Firmware e di Fare Elettronica in un comodo file zip da scaricare! In PDF ad alta risoluzione tutti i numeri della rivista Firmware dal 2006 al 2011 e Fare Elettronica dal 2003 al 2011.

**CODICE: FEFW-DVD PREZZO: € 30,24**

## Modulo PT100 per MOACON

Modulo input temperatura con PT100 per MOACON. Sensore temperatura PT100Ω a 4 canali -100~500

**CODICE: RS-THRT4 PREZZO: € 120,88**

## Modulo Convertitore AD per MOACON

Modulo convertitore AD ad alta risoluzione: 16,6 bit a 4 canali. 0~10 V, 1~5 V, 4~20 mA

**CODICE: RS-HADIN4 PREZZO: € 157,18**

## Modulo convertitore DA per MOACON.

MOACON è un controllore industriale modulare programmabile in C. Grazie al suo design modulare, il MOACON consente ai progettisti di adottare solo i moduli necessari per i requisiti specifici di progetto, permettendo una ottimale adattabilità alle esigenze specifiche. Il MOACON grazie alla CPU ARM a 32 bit permette elevate prestazioni. Sono disponibili singoli moduli con funzionalità di I/O digitali, uscita a relè, conversione analogico-digitale e digitale-analogico, controllo motore, controllo della temperatura, RS-232 e Ethernet, e potenzialmente molti altri. L'ambiente di sviluppo integrato e gratuito, il MOACON Studio, dispone di un compilatore C, editor, RS-232, il download e il debug tramite USB e altre funzionalità. L'editor permette l'evidenziazione della sintassi, il completamento automatico dei comandi, help sensibile al contesto, tutto per rendere l'apprendimento, lo sviluppo, il test dei progetti con MOACON ad elevata produttività oltre ad essere una piacevole esperienza. RS-DAOUT2B è un modulo convertitore DA con uscita in corrente a 16 bit a 2 canali.

**CODICE: RS-DAOUT2B PREZZO: € 120,88**

## prossimamente su Fare Elettronica

COMING...

### DEMOBOARD PER RASPBERRY PI

Come realizzare DEMO BOARD con cui poterci esercitare pilotando le uscite o acquisendo gli ingressi. Nei prossimi numeri saranno analizzati i segnali presenti sul connettore della scheda Raspberry Pi e saranno presentati alcuni programmi per l'utilizzo della scheda Demo.

### LUCI MUSICALI VIA USB

Come costruire una scheda che permette di creare un'infinità di giochi di luci personalizzati. La scheda, già alimentata direttamente da 230V, comprende: LED monitor dei giochi, uscite di potenza da 300 W per canale e isolamento optoelettronico. Un facile programma in grado di gestire i giochi anche al ritmo della musica riprodotta dal PC stesso, rende tutto, oltre che divertente, estremamente facile e professionale.

### ESPERIMENTI CON IL PWM

In questo numero abbiamo illustrato la tecnica base per la generazione dei segnali PWM. Nel prossimo numero vedremo come utilizzare il PWM via hardware e vedremo alcune applicazioni pratiche semplici ma molto interessanti basate appunto sulla modulazione a larghezza di impulso.

**IMPORTANTE: NON SCRIVERE NELLA ZONA SOPRASTANTE!**

AVVERTENZE

Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.

La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.

Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

PRODOTTI

Se vuoi ordinare "offline" ovvero senza usare Internet, usa il seguente modulo d'ordine **PRODOTTI**

| COD. PRODOTTO | DESCRIZIONE | PREZZO UNITARIO | Q.tà | Totale |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | SUB TOTALE | |
| | | | SPESE DI SPEDIZIONE | |
| | | | TOTALE | |

*Nota. Se le righe non sono sufficienti, utilizzare più copie del modulo.*
*Per maggiori dettagli sulle spese di spedizione e i metodi di pagamento vedi retro cartolina.*

**METODI DI PAGAMENTO**

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **CONTRASSEGNO**

☐ MasterCard ☐ VISA ☐ postepay

TITOLARE. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
NUMERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
DATA DI SCADENZA. . . . . . . . . . . . . . . . . . . . . . . . . CODICE DI SICUREZZA . . . . . . . . . . . .

FE 329

**INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225**
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . . . . . . . . . . . . . . . . . . . . . . . . Cognome . . . . . . . . . . . . . . . . . . . . . . . . . .
Indirizzo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . n° . . . . . . . . . .
Cap . . . . . . . . . . . Città . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Prov . . . . . . . .
Tel . . . . . . . . . . . . . . . . . . . . . Fax . . . . . . . . . . . . . . . . . . . . Email . . . . . . . . . . . . . . . . . . . .
Ragione Sociale . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Codice fiscale . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Partita IVA . . . . . . . . . . . . . . . . . . . . . .

DATA . . . . . . . . . . . . . . . . . . . . . . . . . . . FIRMA . . . . . . . . . . . . . . . . . . . . . . . . . . . . ☐ Fattura

ABBONAMENTO

## *Sì, mi abbono a Fare Elettronica*

☐ Abbonamento TRIAL a 3 numeri di FARE ELETTRONICA a soli € 16,50 anziché € 18,00
☐ Abbonamento PREMIUM a 11 numeri di FARE ELETTRONICA a soli € 55,00 anziché € 66,00
☐ Abbonamento PRO a 11 numeri di FARE ELETTRONICA include il CD dell'annata 2011 a soli € 59,50 anzichè € 96,00

FE 329

**METODI DI PAGAMENTO** (per maggiori dettagli vedi retro cartolina)

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
NUMERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
DATA DI SCADENZA. . . . . . . . . . . . . . . . . . . . . . . . . CODICE DI SICUREZZA . . . . . . . . . . . .

**INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225**
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . . . . . . . . . . . . . . . . . . . . . . . . Cognome . . . . . . . . . . . . . . . . . . . . . . . . . .
Indirizzo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . n° . . . . . . . . . .
Cap . . . . . . . . . . . Città . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Prov . . . . . . . .
Tel . . . . . . . . . . . . . . . . . . . . . Fax . . . . . . . . . . . . . . . . . . . . Email . . . . . . . . . . . . . . . . . . . .
Ragione Sociale . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Codice fiscale . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Partita IVA . . . . . . . . . . . . . . . . . . . . . .

DATA . . . . . . . . . . . . . . . . . . . . . . . . . . . FIRMA . . . . . . . . . . . . . . . . . . . . . . . . . . . . ☐ Fattura

## ISTRUZIONI PER L'ORDINE E SPESE DI SPEDIZIONE

Il modulo d'ordine dovrà essere compilato in tutte le sue parti ed inviato via posta o via fax ai recapiti indicati sul modulo d'ordine stesso. Gli ordini potranno essere fatti anche direttamente online dal sito www.elettroshop.com/ieshop. In questo caso non sarà necessario inviare il modulo d'ordine. Le spese di spedizione ammontano a euro 8,50 a cui vanno aggiunti euro 3,50 se si sceglie di pagare in contrassegno.

## METODI DI PAGAMENTO

Si accettano pagamenti ino in contrassegno, carta di credit, bollettino postale o bonifico all'ordine. Per il contrassegno verrà applicata una spesa aggiuntiva di euro 3,50 per le spese di contrassegno. Forme diverse di pagamento devono essere previamente concordate.

### Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

### Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento.

### Contrassegno

La merce verrà pagata direttamente al corriere alla consegna della merce. Il pagamento in contrassegno comporta l'addebito di euro 3,50 per spese di contrassegno.

### Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

## TERMINI E CONDIZIONI

Tutti i prodotti sono coperti da garanzia. La garanzia copre tutti i difetti di fabbricazione entro un anno dal ricevimento della merce. Tutti i prodotti non funzionanti per uso improprio o incuria non saranno ritenuti in garanzia e saranno addebitati gli eventuali costi di riparazione. Tutti i prodotti verranno riparati e/o sostituiti direttamente dal produttore. Non sono coperti da garanzia i componenti elettronici (microprocessori, memorie, ecc.) La garanzia dei prodotti si intende F.co ns. sede, le eventuali spese di trasporto sono a carico del cliente salvo accordi diversi. Per dar corso alla riparazione/sostituzione in garanzia è necessario seguire l'apposita procedura di RMA.

## PRIVACY

Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni srl, Via Giotto 7 – 20032 Cormano o tramite email all'indirizzo info@inwaredizioni.it

---

## ABBONARSI ALLE RIVISTE INWARE EDIZIONI CONVIENE!

I vantaggi per gli abbonati sono, oltre al prezzo bloccato per un anno, la ricezione del numero direttamente a casa con la garanzia di ricevere tutti i numeri. Inoltre un vistoso risparmio che, nel caso dell'abbonamento PLUS, ammonta a ben 36,50 euro. L'abbonamento ha una durata di 12 mesi e comporta l'invio di 11 numeri di Fare Elettronica. Eventuali variazioni di indirizzo andranno comunicate tempestivamente alla Redazione che provvederà a registrare il cambiamento senza alcuna spesa aggiuntiva. L'abbonamento decorrerà dal primo numero raggiungibile alla data di avvenuto pagamento. Non sono previsti rimborsi in caso di disdetta dell'abbonamento.

## METODI DI PAGAMENTO

### Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

### Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento indicando nella causale: "Abbonamento Fare Elettronica"

### Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

### Assegno bancario

È possibile spedire un assegno bancario insieme a questo coupon.

### Paypal

Inviare il pagamento a paypal@invaredizioni.it.